

AD-A088 422

PURDUE UNIV LAFAYETTE IND
PRANC: PROGRAM FOR ANALYZING NONLINEAR CIRCUITS.(U)
MAY 80 H K THAPAR, B J LEON

F/G 9/3

F30602-78-C-0102

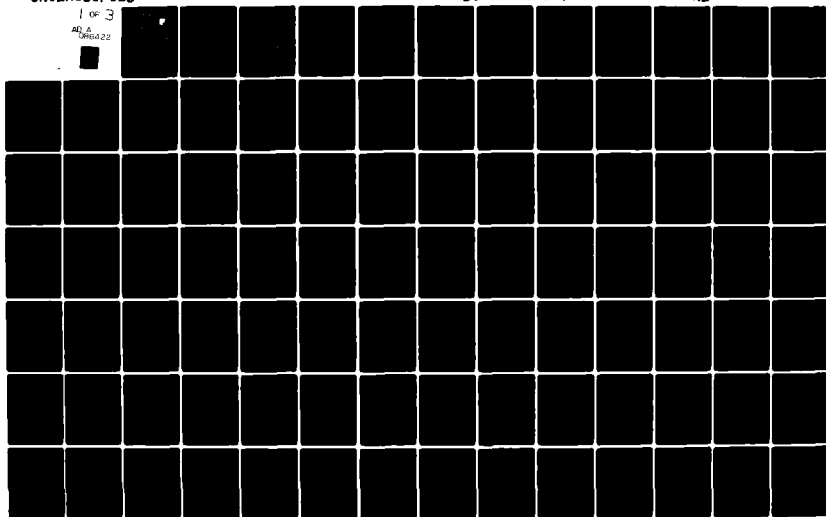
UNCLASSIFIED

RADC-TR-80-139

NL

1 of 3

AD-A088422



52
RADC-TR-80-139

Phase Report

May 1980

LEVEL

12

AD A088422



PRANC: PROGRAM FOR ANALYZING NONLINEAR CIRCUITS

Purdue University

H. K. Thapar

B. J. Leon

**DTIC
ELECTE
AUG 26 1980**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

**ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441**

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-80-139 has been reviewed and is approved for publication.

APPROVED:

Jon B. Valente
JON B. VALENTE
Project Engineer

APPROVED:

David C. Luke

DAVID C. LUKE, Lt Colonel, USAF
Chief, Reliability and Compatibility Division

FOR THE COMMANDER:

John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBCT) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-80-139	2. GOVT ACCESSION NO. AD-A088	3. RECIPIENT'S CATALOG NUMBER 422
4. TITLE (and Subtitle) PRANC: PROGRAM FOR ANALYZING NONLINEAR CIRCUITS	5. TYPE OF REPORT & PERIOD COVERED Phase Report Nov 78 - Nov 79	6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) H. K./Thapar B. J./Leon	8. CONTRACT OR GRANT NUMBER(s) F30602-78-C-0102	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Purdue University West Lafayette IN 47907	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 233803PE	
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBCT) Griffiss AFB NY 13441	12. REPORT DATE May 1980	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same	13. NUMBER OF PAGES 232	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Jon B. Valente (RBCT)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Sinusoidal Steady-state; Functional Analysis; Computer-aided Analysis; Multi-dimensional Transforms		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The application of the Volterra series method for the analysis of circuits with multiple nonlinear elements and multiple multi-frequency input sources is described. Using multi-dimensional transform theory, an approach for deriving the Volterra transfer functions is given. The computation of the sinusoidal steady-state response from these transfer functions is presented.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Finally, algorithms for adapting the Volterra series method for computer-aided steady-state analysis of nonlinear circuits are described. A complete documentation of the program PRANC, which uses the Volterra series approach, is also contained in this report.

Accession For	
NTIS	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	
Justification	
By	
Distribution/	
Availability	
Dist	Availability
A	

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This effort was conducted by Purdue University under the sponsorship of the Rome Air Development Center Post-Doctoral Program. Mr. Jon Valente of RADC was the task project engineer and provided overall technical direction and guidance. Prof. B. J. Leon directed this research and the preparation of this report at Purdue University. The authors of the report are B. J. Leon and H. K. Thapar.

The RADC Post-Doctoral Program is a cooperative venture between RADC and some sixty-five universities eligible to participate in the program. Syracuse University, Clarkson College of Technology and the Georgia Institute of Technology act as prime contractor schools with other schools participating via sub-contracts with the prime schools. The U.S. Air Force Academy (Dept. of Electrical Engineering), Air Force Institute of Technology (Dept. of Electrical Engineering) and the Naval Post Graduate School (Dept. of Electrical Engineering) also participate in the program.

The Post-Doctoral Program provides an opportunity for faculty at participating universities to spend up to one year full time on exploratory development and problem-solving efforts with the post-doctorals splitting their time between the customer location and their educational institutions.

Further information about the RADC Post-Doctoral Program can be obtained from Jacob Scherer, RADC/RBC, Griffiss AFB, NY, 13441, telephone AV 587-2543, COMM (315) 330-2543.

This document is the final report for Task 7 of Purdue University's Sub-contract from Clarkson College of Technology. The task was to "Develop and Apply Symbolic Methods to the Volterra Series Approach to Nonlinear Circuit Analysis."

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER 1 - INTRODUCTION	1
1-1. Statement of Objectives	1
1-2. Organization of the Report	2
CHAPTER 2 - VOLTERRA SERIES METHOD	3
2-1. Introduction	3
2-2. Multi-dimensional Transforms	3
2-3. A Nonlinear Differential Equation	9
2-4. Multiple-Node, Multiple-Nonlinearity Circuit Analysis	15
2-5. Multiple Input Circuit Analysis	29
2-6. Sinusoidal Steady-State Analysis	34
CHAPTER 3 - COMPUTER-AIDED ANALYSIS USING VOLTERRA SERIES	42
3-1. Introduction	42
3-2. Why a Symbolic Approach	43
3-3. Symbolic Analysis Method	46
3-4. Spectrum and Distortion Analysis Algorithm	53
3-5. Program PRANC	56
CHAPTER 4 - USER'S GUIDE FOR PRANC	61
4-1. Introduction	61
4-2. Preliminary Data Preparation	62
4-3. Input Description for PRANC	73
4-4. Interpretation of PRANC output	87
4-5. Examples using PRANC	90
CHAPTER 5 - PROGRAMMER'S GUIDE FOR PRANC	140
5-1. Introduction	140
5-2. Program Structure of PRANC	141
5-3. Glossary and Sub-program listings for PRANC	147
5-4. System Dependent Cards	225
CHAPTER 6 - CONCLUDING REMARKS	226
REFERENCES	228
APPENDIX	230

LIST OF TABLES

Table	Page
2-1. Nonlinear current sources in multiple-node, multiple-nonlinearity circuit analysis	24,25
4-1. Summary of available options using PRANC	77
5-1. List of system dependent cards	225

LIST OF FIGURES

Figure	Page
2-1. Steps in nonlinear circuit analysis using volterra series	19,20
2-2. Determination of Volterra transfer functions	28
2-3. Multiple input nonlinear circuit analysis	31,32
3-1. Determination of $Z(s)$ for the p-port network using state equations	49
3-2. Algorithm for inverting $Y(s)$ symbolically	52
3-3. Algorithm for distortion and Spectrum Analysis	54
4-1. PRANC element definitions	63,64
4-2. Transistor equivalent model	67
4-3. Sequence of cards for single input circuits	74
4-4. Sequence of cards for two input circuits	75
4-5. A transistor amplifier circuit	91
4-6. Data preparation for Example 4-1	93
4-7. Data cards for Example 4-1	94
4-8. Circuit diagram for Example 4-2	115
4-9. Data cards for Example 4-2	116
5-1. Program structure of PRANC	143

CHAPTER 1

INTRODUCTION

1-1. Statement of the Objectives

In the analysis of nonlinear systems, two main classes of solutions are generally sought: 1) transient, and 2) steady state. The basic goal of this investigation is to obtain the sinusoidal steady-state solution of nonlinear circuits via the Volterra series method [1-14].

The most commonly used present-day approach for analyzing nonlinear systems is numerical integration [20]. The nonlinear differential equations are integrated from some initial time, t_0 , to some final time, t_f . When the sinusoidal steady-state response is sought, the value of t_f chosen is usually large to insure that all transients have been eliminated. A subsequent fast Fourier analysis yields the frequency components of the output response. A more efficient method for obtaining the sinusoidal steady-state response is to pose the analysis problem as a two-point boundary value problem and then apply Newton's method [20]. This approach, however, allows for only single frequency inputs.

The problems involved in the numerical integration method are well known [20]. These problems notwithstanding, there are other inefficiencies. When one is solely interested in the steady-state response, the computation expended in reaching t_f is a waste. This inefficiency grows as the poles of the linearized system move close to the imaginary axis, as is often the case in many quasi-linear communication circuits.

Other methods such as the harmonic balance or the describing function method are seldom used, simply because the assumption behind these methods render them undependable. The Picard iteration method [14] is often used in nonlinear systems analysis. This method also has limitations when used for

computer-aided analysis, particularly when multi-tone inputs are present.

The fundamental intent behind this report is to examine the computational aspect of the Volterra series when used for the steady-state analysis of circuits with multiple nonlinearities and multiple multi-tone input sources. A basic algorithm for adapting this method for computer-aided analysis is developed. Its implementation as a digital computer program, entitled PRANC (Program for Analyzing Nonlinear Circuits), is also included in this report.

1-2. Organization of the Report

After this introductory chapter, this report contains the following five chapters.

Chapter 2, entitled "Volterra Series Method", discusses the analysis method which forms the basis of this investigation. A systematic approach for system characterization in the transform domain is developed. The determination of the sinusoidal steady-state response for multi-tone inputs from the system characterization is also developed.

Chapter 3 considers the computational aspect of the Volterra series method. An algorithm, which uses semi-symbolic analysis [20], is developed for the efficient implementation of this method on a digital computer. An overview of PRANC is also presented in this chapter.

Chapter 4 provides the User's guide for PRANC. Several examples to illustrate the use of this program are included here.

Chapter 5 contains the Programmer's guide for PRANC. Each sub-program listing, together with its functions, is documented in this chapter.

Finally, Chapter 6 is reserved for some concluding remarks.

CHAPTER 2

VOLTERRA SERIES METHOD

2-1. Introduction

Nonlinear systems that admit a Volterra series description are completely characterized by their nonlinear impulse response functions or the generalized transfer functions, which are the multi-dimensional transforms of the nonlinear impulse response functions. Thus, any analysis of nonlinear systems via the Volterra series method will entail the determination of either one of these functions.

The method for determining the generalized transfer functions given in [13] will be presented here. This method relies on the application of multi-dimensional transforms to a set of differential equations. In section 2-2 the multi-dimensional transform theory is introduced, along with the application of the theory to specific examples which will be subsequently used in deriving the generalized transfer functions. In section 2-3 the generalized transfer functions for an r -th order scalar nonlinear differential equation are obtained. Section 2-4 is devoted to the determination of the nonlinear transfer functions of a general multiple-node, multiple-nonlinearity circuit with a single input. The case of multiple input sources is treated in section 2-5. Section 2-6 shows the relationship between the terms in the sinusoidal steady-state response and the generalized transfer functions.

2-2. Multi-dimensional Transforms

The Laplace transform pair of a one-dimensional function, $f(t)$, is:

$$F(s) = \int_{-\infty}^{\infty} f(t) e^{-st} dt \quad (2-1)$$

and

$$f(t) = \frac{1}{(2\pi j)} \int_{\sigma-j\infty}^{\sigma+j\infty} F(s) e^{st} ds \quad (2-2)$$

For a multi-variable function, $f(t_1, t_2, \dots, t_n)$, the corresponding multi-dimensional transform [15] is:

$$F(s_1, s_2, \dots, s_n) = \int \dots \int_{n\text{-fold}} f(t_1, t_2, \dots, t_n) \exp(-s_1 t_1 - \dots - s_n t_n) dt_1 \dots dt_n \quad (2-3)$$

and

$$f(t_1, \dots, t_n) = \frac{1}{(2\pi j)^n} \int \dots \int_{n\text{-fold}} F(s_1, \dots, s_n) \exp(s_1 t_1 + \dots + s_n t_n) ds_1 \dots ds_n \quad (2-4)$$

$$f(t_1, \dots, t_n) \leftrightarrow F(s_1, \dots, s_n) \quad (2-5)$$

Before proceeding further, we make the following notational definitions:

$$F(s_1, s_2, \dots, s_n) = \mathcal{L}[f(t_1, t_2, \dots, t_n)] \quad (2-6)$$

and

$$f(t_1, t_2, \dots, t_n) = \mathcal{L}^{-1}[F(s_1, s_2, \dots, s_n)] \quad (2-7)$$

Whether we use Fourier transform or Laplace transform in eqns. (2-3) and (2-4) depends on the contours of integration and values of s_1, s_2, \dots, s_n . The importance of the region of convergence when dealing with unstable and non-causal linear systems is well known. Here we assume that the systems under consideration are causal; that is, the Volterra kernels

$h_n(t_1, t_2, \dots, t_n) = 0$, for $t_1, t_2, \dots, t_n \leq 0$. Also, in general, we are concerned with functions (or generalized functions) whose region of convergence includes the imaginary axis in each variable, so that the Fourier transform is included in our definitions.

It should also be noted that most of the properties of the one-dimensional transform (linear case) carry over to the multi-dimensional case. The validity of this statement can be checked elsewhere [5].

It is often desirable to express the multi-variable function, $f(t_1, t_2, \dots, t_n)$, as a simple function of time, $f(t)$, and vice versa. If all t_i 's are restricted to be identical so that $t = t_1 = t_2 = \dots = t_n$, then $f(t_1, t_2, \dots, t_n)$ becomes $f(t)$. Thus, in the two variable case, $f(t)$ can be obtained from $f(t_1, t_2)$ by evaluating $f(t_1, t_2)$ along the 45° line $t_1 = t_2$. Similarly, if we plot $f(t_1, t_2, t_3)$ in a three-dimensional space, then, to obtain $f(t)$, we are only interested in $f(t_1, t_2, t_3)$ along the line $t_1 = t_2 = t_3$. The idea of converting a nonlinear function of one variable t into a product of linear multi-variable functions will be used repeatedly in the sequel. One must, however, bear in mind that the ultimate goal is to obtain the solution of the differential equation as a function of time, t , and that the introduction of t_1, t_2 , etc. are merely for mathematical manipulations.

We now apply multi-dimensional transforms to some specific cases which will be subsequently used in sections (2-3) and (2-4).

2-2.1 Volterra Series: The Volterra series relates the system input $x(t)$ to the system output $y(t)$ as follows*:

*Unless otherwise stated, all limits of integration are between 0 and ∞ in our discussion here.

$$\begin{aligned}
y(t) &= \sum_{n=1}^{\infty} \int \cdots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i \\
&= \sum_{n=1}^{\infty} y_n(t)
\end{aligned} \tag{2-8}$$

where

$$y_n(t) = \int \cdots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i \tag{2-9}$$

Introducing dummy variables t_1, t_2, \dots, t_n in eqn. (2-9) we can write $y_n(t)$ as:

$$\begin{aligned}
y_n(t) &= y_n(t_1, t_2, \dots, t_n) |_{t_1=t_2=\dots=t_n=t} \\
&= \int \cdots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t_i - \tau_i) d\tau_i
\end{aligned} \tag{2-10}$$

Taking the n-dimensional transforms of eqn. (2-10), we get:

$$\begin{aligned}
Y_n(s_1, \dots, s_n) &= \mathcal{L}[y_n(t_1, \dots, t_n)] \\
&= \int \cdots \int_{2n\text{-fold}} h_n(\tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n x(t_i - \tau_i) e^{-s_i t_i} d\tau_i dt_i
\end{aligned} \tag{2-11}$$

Defining $t_n - \tau_n = \sigma_n$, $t_{n-1} - \tau_{n-1} = \sigma_{n-1}$, \dots , $t_1 - \tau_1 = \sigma_1$, and therefore:

$$t_n = \sigma_n + \tau_n, \quad t_{n-1} = \sigma_{n-1} + \tau_{n-1}, \dots, \quad t_1 = \sigma_1 + \tau_1;$$

$d\sigma_n = dt_n$, $d\sigma_{n-1} = dt_{n-1}$, \dots , $d\sigma_1 = dt_1$. Substituting these quantities in eqn. (2-11) and performing the 2n-fold integrations with respect to τ_i and

σ_i gives

$$Y_n(s_1, \dots, s_n) = H_n(s_1, \dots, s_n) \prod_{i=1}^n X(s_i) \quad (2-12)$$

where $H_n(s_1, \dots, s_n)$ and $X(s_i)$ are the transforms of $h_n(t_1, t_2, \dots, t_n)$ and $x(t_i)$ respectively. Therefore the transform domain description of eqn. (2-8) becomes:

$$Y(s_1, s_2, \dots, s_n) = \sum_{n=1}^{\infty} H_n(s_1, \dots, s_n) \prod_{i=1}^n X(s_i) \quad (2-13)$$

If the input $x(t)$ is a delta function, then eqns. (2-12) and (2-13) reduce, respectively, to:

$$Y_n(s_1, \dots, s_n) = H_n(s_1, s_2, \dots, s_n) \quad (2-14)$$

and

$$Y(s_1, \dots, s_n) = \sum_{n=1}^{\infty} H_n(s_1, \dots, s_n) \quad (2-15)$$

Equations (2-12) through (2-14) will be used repeatedly in section (2-3).

2-2.2 Nonlinear Terms. The characteristics of nonlinear elements encountered in many nonlinear dynamical systems can be represented over any finite range by a polynomial. This gives rise to nonlinear differential equations with polynomial type nonlinear terms. When such elements are used in a system, the equilibrium equations contain integrals and derivatives of the polynomials. We can apply multi-dimensional transforms to these nonlinear terms by first converting an nth power to an n-fold product of terms with different domains. More detail on these derivatives is given in [13].

$y^n(t)$ Term: Consider an n -dimensional time space with variables t_i , $i=1,2,\dots,n$. From the single variable function $y(t)$ define an n -variable functional $y(t_1, t_2, \dots, t_n) = \prod_{i=1}^n y(t_i)$. Then

$$y^n(t) = y(t_1, t_2, \dots, t_n) \quad \forall t_i = t \quad (2-16)$$

and

$$Y(s_1, \dots, s_n) = \prod_{i=1}^n Y(s_i) \quad (2-17)$$

$\frac{d}{dt} y^n(t)$ Term:

$$\frac{d}{dt} y^n(t) = \frac{d}{dt} y(t_1, \dots, t_n) \Big|_{t_1=t_2=\dots=t_n=t} \quad (2-18)$$

$$\begin{aligned} Y(s_1, s_2, \dots, s_n) &= \mathcal{L} \left[\sum_{s=1}^n \frac{\partial}{\partial t_s} y(t_1, \dots, t_n) \frac{dt_s}{dt} \right] \\ &= (s_1 + s_2 + \dots + s_n) Y(s_1) \dots Y(s_n) \end{aligned} \quad (2-19)$$

$\int y^n(t) dt$ Term:

$$\int y^n(t) dt = \int y_n(\tau_1 - t, \tau_2 - t, \dots, \tau_n - t) dt \quad (2-20)$$

Letting $\tau_i - t = t_i$, and taking the transform of eqn. (2-20), we get

$$Y(s_1, s_2, \dots, s_n) = Y_n(s_1, \dots, s_n) / (s_1 + s_2 + \dots + s_n) \quad (2-21)$$

$$= \left[\frac{1}{s_1 + s_2 + \dots + s_n} \right] \prod_{i=1}^n Y(s_i) \quad (2-22)$$

The general forms in eqns. (2-12), (2-19) and (2-27) will be used in sections (2-3) and (2-4). The salient feature in each of these equations is how an nth degree polynomial function in the time-domain is represented by the nth-order product of the transform of the function in the transform domain. It is this product structure which, analogous to the case of linear system analysis, makes the analysis of nonlinear systems easier via the transform-domain approach.

2-3. A Nonlinear Differential Equation:

In this section, we present a method, based on applying the multi-dimensional transforms to nonlinear differential equations, to determine the response of a nonlinear system with a functional power series type of non-linearity. The nonlinear differential equation considered is the following:

$$L_1[y(t)] + L_2\left[\sum_{n=2}^N a_n y^n(t)\right] = x(t) \quad (2-23)$$

where $x(t)$ and $y(t)$ are system input and output, respectively, L_1 is a linear differential operator:

$$L_1[\cdot] = \sum_{r=0}^R \frac{d^r}{dt^r} [\cdot] \quad (2-24)$$

and L_2 is $\frac{d}{dt}$, \int , or a constant, or a sum of these operators. It should be noted that the linear operator, L_2 , operates on a polynomial function of $y(t)$.

We now present an approach whereby the nonlinear differential equation (2-23) is solved by a bootstrapping operation by first dissolving it into a set of linear differential equations with nonlinear inputs. Multidimensional transforms are then applied to these new equations to obtain the Volterra series solution.

There are many different methods of rendering a nonlinear differential equation into a sequence of linear differential equations involving successively higher order outputs with known nonlinear input terms. We use the approach outlined in [12].

Assume that the input in eqn. (2-23) is of the form

$$x(t) = \epsilon v(t) \quad (2-25)$$

The dummy variable ϵ helps to keep track of the order of the terms: a term with coefficient ϵ^n signifies an n th order term. This can be seen easily by substituting eqn. (2-25) in eqn. (2-9), which yields:

$$y_n(t) = \epsilon^n \int \dots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n v(t-\tau_i) d\tau_i \quad (2-26)$$

Let us assume that $r(t)$ is the response to the input $v(t)$ in eqn (2-23). Then, according to the Volterra series expansion, as per eqn. (2-8) and (2-9), the n -th order response is:

$$r_n(t) = \int \dots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n v(t-\tau_i) d\tau_i \quad (2-27)$$

Comparing (2-27) and (2-26), we obtain the following relationships:

$$y_n(t) = \epsilon^n r_n(t) \quad (2-28)$$

and therefore, as per eqn. (2-8),

$$y(t) = \sum_{n=1}^{\infty} y_n(t) = \sum_{n=1}^{\infty} \epsilon^n r_n(t) \quad (2-29)$$

We now have two differential equations which relate $r(t)$ and $v(t)$. First, equation (2-23) can be re-written as:

$$L_1[r(t)] + L_2\left[\sum_{n=2}^N a_n r^n(t)\right] = v(t) \quad (2-30)$$

Second, after substituting eqn. (2-29) into (2-23), we get:

$$L_1\left[\sum_{n=1}^{\infty} \epsilon^n r_n(t)\right] + L_2\left[\sum_{j=2}^N \left(a_j \sum_{n=1}^{\infty} \epsilon^n r_n(t)\right)^j\right] = \epsilon v(t) \quad (2-31)$$

Thus in order to solve eqn. (2-23), we can solve eqn. (2-31) for $r_n(t)$, $n = 1, 2, \dots$ and substitute in eqn. (2-29) to solve for $y(t)$ after setting $\epsilon = 1$. Setting $\epsilon = 1$ implies that $x(t) = v(t)$, and therefore $y(t) = r(t) = \sum_n r_n(t)$. The introduction of ϵ is a mathematical artifice which helps to equate coefficients of ϵ^n on both sides of eqn. (2-31), thereby yielding linear differential equations (involving successively higher order outputs) with non-linear inputs.

To solve for $r_1(t)$, the linear system response, we equate coefficients of ϵ^1 on both sides of eqn. (2-31), thus yielding the following equation:

$$L_1[r_1(t)] = v(t) \quad (2-32)$$

Similarly we equate coefficients of $\epsilon^2, \epsilon^3, \epsilon^4, \epsilon^5$, and so on, on both sides of eqn. (2-31) to obtain the following equations:

$$L_1[r_2(t)] + L_2[a_2 r_1^2(t)] = 0 \quad (2-33)$$

$$L_1[r_3(t)] + L_2[2a_2 r_1(t)r_2(t) + a_3 r_1^3(t)] = 0 \quad (2-34)$$

$$L_1[r_4(t)] + L_2[a_2(2r_1(t)r_3(t) + r_2^2(t)) + 3a_3 r_1^2(t)r_2(t) + a_4 r_1^4(t)] = 0 \quad (2-35)$$

$$L_1[r_5(t)] + L_2[2a_2 r_1(t)r_4(t) + a_3(3r_1^2(t)r_3(t) + 3r_1(t)r_2^2(t)) + 4a_4 r_1^3(t)r_2(t) + a_5 r_1^5(t)] = 0 \quad (2-36)$$

.

.

.

To solve for the generalized transfer functions of eqn. (2-30), we take the 1-dimensional transform of eqn. (2-32) and obtain:

$$L_1(s_1)R_1(s_1) = V(s_1) \quad (2-37)$$

If $v(t) = \delta(t)$, then $V(s_1) = 1$, and therefore, according to eqn. (2-14), we have

$$R_1(s_1) = H_1(s_1) = \frac{1}{L_1(s_1)} \quad (2-38)$$

To solve for the second-order transfer functions, $H_2(s_1, s_2)$, we extend the second term of eqn. (2-33) to a two dimensional domain. Since the physical system is not defined when $t_1 \neq t_2$ we can assume that the extension of eqn. (2-33) holds for all t_1 and t_2 . Transforming via eqn. (2-17) gives

$$L_1(s_1+s_2)R_2(s_1,s_2) + a_2L_2(s_1+s_2)R_1(s_1)R_1(s_2) = 0 \quad (2-39)$$

Using (2-14) and (2-38) in eqn. (2-39), we obtain

$$R_2(s_1,s_2) = H_2(s_1,s_2) = - \frac{a_2L_2(s_1+s_2)H_1(s_1)H_1(s_2)}{L_1(s_1+s_2)} \quad (2-40)$$

For R_3 and higher terms we find that the order of variables t_1, t_2, t_3 seems important. Physically this should not be. We can symmetrize by averaging. That is, we sum each of the n th order transfer function over all permutations of its arguments and divide by the number of components in the sum. We use an overbar to represent the symmetrized function.

$$\begin{aligned} L_1(s_1+s_2+s_3)R_3(s_1,s_2,s_3) + L_2(s_1+s_2)[2a_2R_1(s_1)R_2(s_2,s_3) \\ + a_3R_1(s_1)R_1(s_2)R_1(s_3)] = 0 \end{aligned} \quad (2-41)$$

Again, using eqns. (2-38), (2-40), and (2-14), we get

$$\begin{aligned} R_3(s_1,s_2,s_3) = H_3(s_1,s_2,s_3) = - \frac{L_2(s_1+s_2+s_3)[2a_2H_1(s_1)H_2(s_2,s_3) \\ + a_3H_1(s_1)H_1(s_2)H_1(s_3)]}{L_1(s_1+s_2+s_3)} \end{aligned} \quad (2-42)$$

In a similar manner, we can derive by inspection:

$$\begin{aligned} H_4(s_1,s_2,s_3,s_4) = - \frac{L_2(\sum_{i=1}^4 s_i)[a_2(2H_1(s_1)H_3(s_2,s_3,s_4) \\ + H_2(s_1,s_2)H_2(s_3,s_4)) + 3a_3H_1(s_1)H_1(s_2)H_2(s_3,s_4)]}{L_1(s_1+s_2+s_3+s_4)} \end{aligned}$$

$$+ a_4 \sum_{i=1}^4 H_1(s_i)] / L_1(\sum_{i=1}^4 s_i) \quad (2-43)$$

and

$$\begin{aligned} H_5(s_1, s_2, s_3, s_4, s_5) = & -L_2\left(\sum_{i=1}^5 s_i\right) \frac{2a_2 H_1(s_1) H_4(s_2, s_3, s_4, s_5)}{+ 3a_3 (H_1(s_1) H_1(s_2) H_3(s_3, s_4, s_5))} \\ & + \frac{H_1(s_1) H_2(s_2, s_3) H_2(s_4, s_5)}{+ 4a_4 H_1(s_1) H_1(s_2) H_1(s_3) H_2(s_4, s_5)} + a_5 \sum_{i=1}^5 H_1(s_i)] / L_1\left(\sum_{i=1}^5 s_i\right) \quad (2-44) \end{aligned}$$

The use of symmetric transfer functions is not merely for notational convenience, but is necessitated by the method we use for introducing the parameters t_1, t_2, \dots , before taking the transforms. Consider a third order term $v_3(t)$ formed as the product of a first order term $v_1(t)$ and a second order $v_2(t)$. On the three dimensional (t_1, t_2, t_3) we could write $v_3(t_1, t_2, t_3)$ as $v_1(t_1)v_2(t_2, t_3)$, $v_1(t_2)v_2(t_1, t_3)$, or $v_1(t_3)v_2(t_1, t_2)$. The first term has transform: $V_1(s_1)V_2(s_2, s_3)$; the second term has: $V_1(s_2)V_2(s_1, s_3)$; and the third has transform: $V_1(s_3)V_2(s_1, s_2)$. When $V_2(\cdot, \cdot)$ is not symmetrical in its arguments, each transformed quantity above will yield a different value. Thus, it becomes necessary to use symmetric transfer functions when performing numerical computations to obtain the system response. It can be shown that the response is unchanged when symmetrized transfer functions are used. Since, in the final analysis we want the value of $v_3(t_1, t_2, t_3)$ only when $t_1 = t_2 = t_3$, we may write

$v_1(t)v_2(t) = \frac{1}{3} [v_1(t_1)v_2(t_2,t_3) + v_1(t_2)v_2(t_1,t_3) + v_1(t_3)v_2(t_1,t_2)]$. This does not change the contribution due to $v_1(t)v_2(t)$ in the system response. In the remaining part of this report we will assume the generalized transfer functions to be symmetric in their arguments.

To conclude this sub-section, we summarize the approach for obtaining the generalized transfer functions of a nonlinear system and also comment on the important ramification of the method. By introducing a dummy variable in the nonlinear differential equation characterizing the system, a set of differential equations of the following form was obtained:

$$L[r_n(t)] + f(r_{n-1}(t)) = 0, \quad n = 2, 3, \dots \quad (2-45)$$

where L is the linear system operator and $f(\cdot)$ is a nonlinear function of $r_{n-1}(t), r_{n-2}(t), \dots, r_1(t)$. $r_1(t)$ is the first-order response, which is simply the response of the linear system. The relationship in eqn. (2-45) is clearly a recursive one, and can be used to solve for $r_n(t)$ in terms of $r_{n-1}(t), r_{n-2}(t)$, etc. This is done by first finding the n -dimensional transform of $f(r_{n-1}(t))$ as discussed above. We then use the transform of eqn. (2-44) to solve for $R_n(s_1, \dots, s_n)$, the n th-order transfer function when the input $v(t)$ is an impulse. The transform of $f(\cdot)$ is done by inspection with the help of the results of section (2-2). The n -dimensional transform of $L[r_n(t)]$ is shown to be $L(s_1 + s_2 + \dots + s_n)R_n(s_1, s_2, \dots, s_n)$. With all this information, eqn. (2-45) is easily solved for the generalized transfer functions.

2-4. Multiple-Node, Multiple-Nonlinearity Circuit Analysis

Many analysis and design problems in circuits and systems involve one or at most a few nonlinear elements in an otherwise linear time-invariant circuit or system. When a single nonlinear element is present, the dif-

ferential equation (2-23) and the material of section (2-2) will be adequate for analyzing the nonlinear circuit. For, in such a case, the linear circuit can be characterized by a convolution kernel (via the Thevenin or Norton Theorems) to give the overall Volterra integral equation [14], which can also be cast in a differential equation form, similar to eqn. (2-23).

However, when multiple nonlinear elements are imbedded in an otherwise linear time-invariant circuit, the analysis entails the solution of a system of nonlinear differential equations. The approach developed in section (2-2) for the scalar case is still applicable, but must be extended to solve the system of nonlinear differential equations.

The number of equations to be solved depends on the number and the type of nonlinear elements considered. When only independent type nonlinear elements are considered, the number of equations is less than or equal to the number of nonlinear elements (assuming that the output is across one of the nonlinear elements; otherwise, an extra equation relating the nonlinear element voltages (currents) and the output voltage (current) is needed to solve for the output). The nonlinear differential equations in such a case is again derived by obtaining the Thevenin (Norton) equivalent circuit (for the linear part of the nonlinear circuit) at each of the ports at which the nonlinear elements are present. When dependent type nonlinear elements are also allowed, the analysis becomes more complicated; for, in such a case, the controlling variables, which may be across a linear element, must be solved for and substituted in the differential equation for the nonlinear element.

Previous works [7,10-12] for determining the generalized voltage ratio transfer functions of lumped nonlinear circuits have applied the harmonic input method, to the nodal analysis. Our discussion in this section for

solving multiple-node, multiple-nonlinearity circuits will be centered around the application of multi-dimensional transforms to a cutset type analysis. Thus, we will be solving for the generalized voltage ratio transfer functions. As we proceed with our discussion, it will become apparent that a cutset analysis approach is the most natural way of solving for the generalized voltage-ratio transfer functions. We now develop the procedure.

The first step in the analysis is to represent each nonlinear element by a polynomial expansion. Thus, in the distortion analysis of transistor amplifiers [7], the exponential type controlled sources in the Ebers-Moll model are first represented by a Taylor series expansion of the function about the quiescent point, thereby yielding a polynomial in terms of the incremental variables. The types of nonlinear elements, and their series representation, that are commonly encountered are:

1. No memory, independent nonlinearity (Nonlinear Resistor)

$$i = F(v) = \sum_{j=1}^{\infty} a_j v^j \quad (2-46)$$

2. No memory, dependent nonlinearity

$$i = G(u, v) = \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} a_{jk} u^j v^k, \quad a_{00} = 0 \quad (2-47)$$

3. Capacitive, independent nonlinearity

$$i = \frac{d}{dt} Q(v) = \frac{d}{dt} \sum_{j=1}^{\infty} a_j v^j \quad (2-48)$$

4. Inductive, independent nonlinearity

$$i = \int_{-\infty}^t \phi(v) dt = \int_{-\infty}^t \sum_{j=1}^{\infty} a_j v^j dt \quad (2-49)$$

where

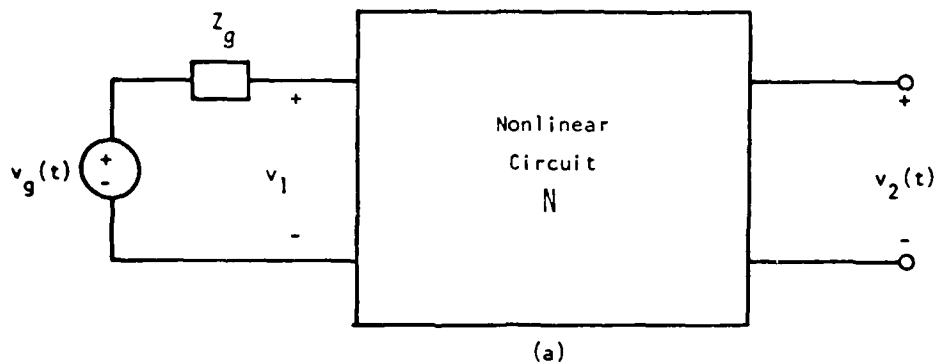
$i \equiv$ incremental current through the element

$v \equiv$ incremental controlling voltage

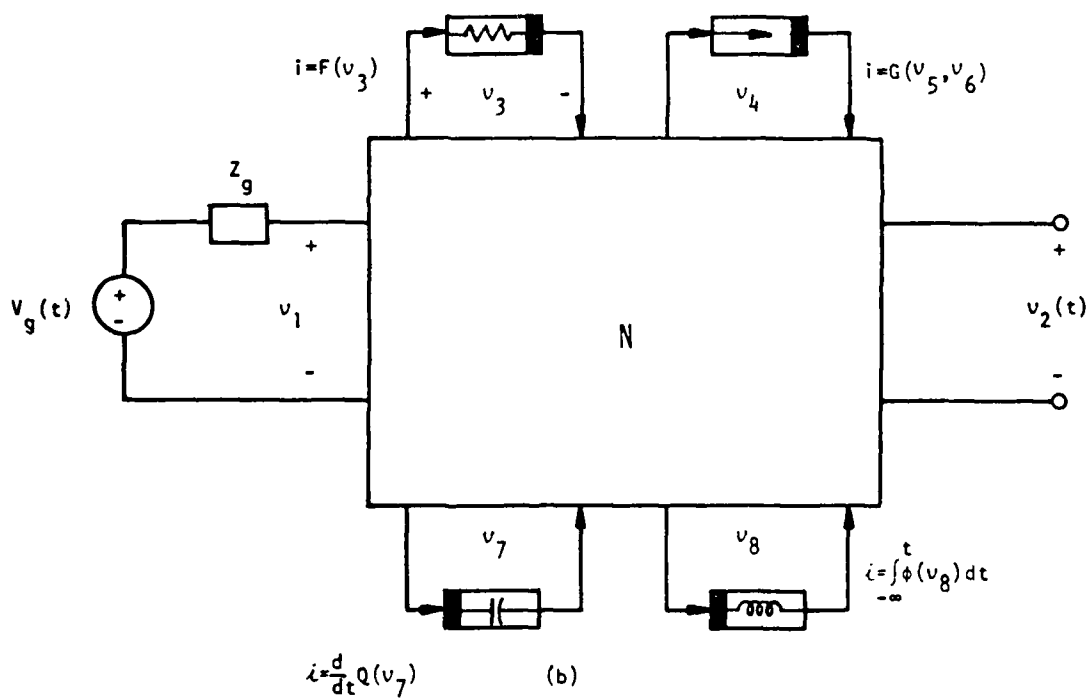
$u \equiv$ incremental controlling voltage

The general procedure employed to solve for the nonlinear transfer functions of a single-input, single-output nonlinear circuit using the cutset analysis approach is illustrated in Fig. 2-1 by considering each of the four nonlinear element types mentioned above.

Consider the nonlinear circuit N, shown in Fig. 2-1(a), containing a nonlinear resistor, a nonlinear dependent source, a nonlinear capacitor, and a nonlinear inductor, where each nonlinear element is voltage controlled. The procedure begins by identifying all the nonlinear elements, as shown in Fig. 2-1(b). We note that the four nonlinear elements depend on six voltages. The next step is to lump the linear parts of the nonlinear elements with the existing linear network to form the augmented linear network. The square, cubic, quartic, etc. terms of the nonlinearity are treated as nonlinear current sources, indicated by i_k^n , meaning the n th order current source at port k . Since the dependent source, $g(v_5, v_6)$, depends on voltages v_5 and v_6 , we also extract these as ports. Thus, altogether we end up with an 8-port linear network, as shown in Fig. 2-1(c).



(a)



(b)

Figure 2-1. Steps in Nonlinear Circuit Analysis using Volterra Series.

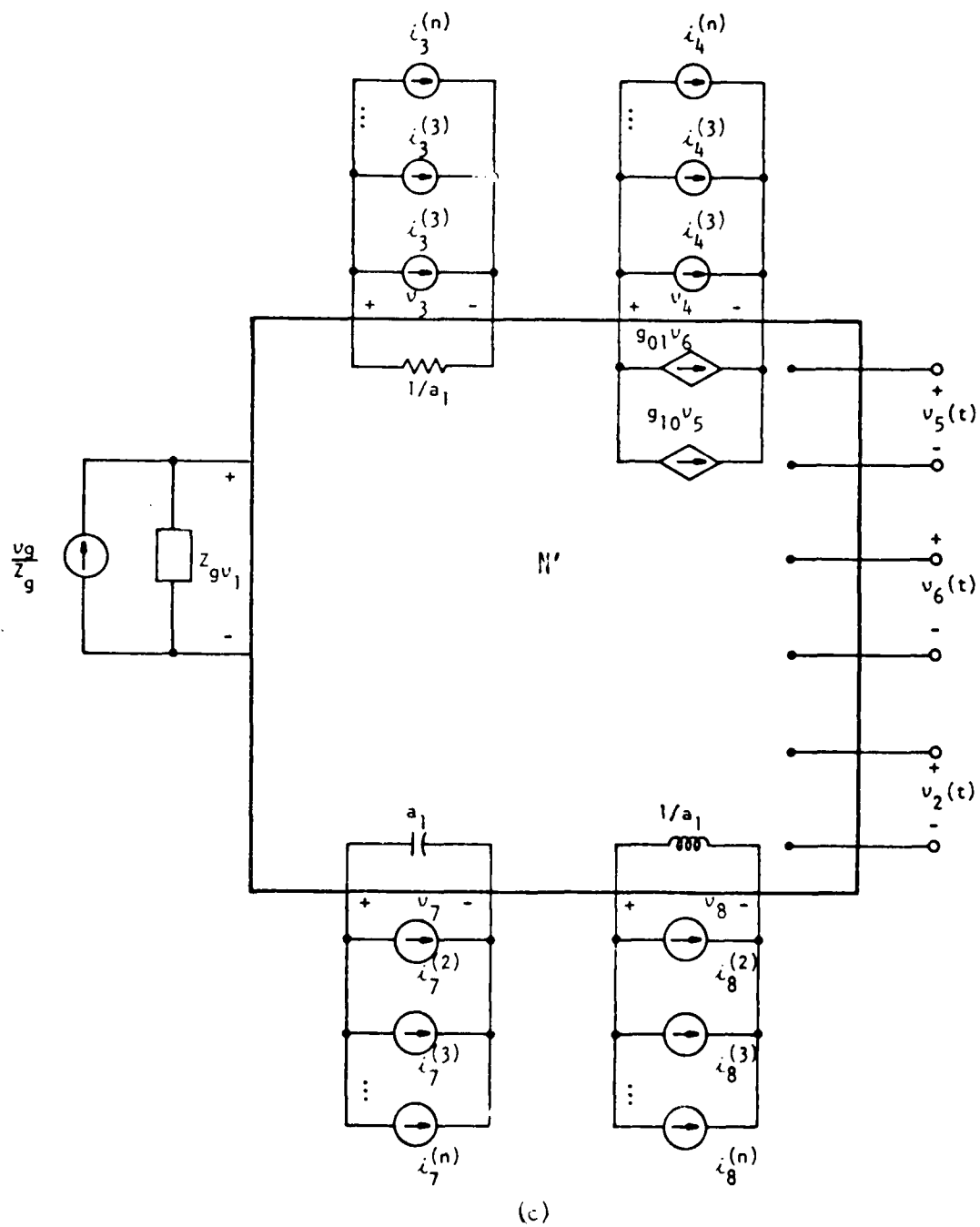


Figure 2-1. (contd.)

The output variables to be found are the voltages at these eight ports. The augmented linear network is denoted by N' in Fig. 2-1(c). To solve for the voltage vector $\underline{v} = [v_1 \ v_2 \ v_3 \ \dots \ v_8]$, we immediately recognize that the branches across these voltage variables must be selected as part of the tree [20]. Clearly, some of the other branches in the augmented linear network may also appear as part of the tree. These will then appear as voltage variables in the cutset equations for the augmented linear network. Since there is no need for these additional variables, we can reduce the dimensionality of our equations by a systematic elimination of these unwanted variables. In the case under consideration, we should be left with only the vector $\underline{v} = [v_1 \ v_2 \ \dots \ v_8]$ as the unknown vector. Each of these 8 ports will have a set of transfer functions of order 1 to n associated with it. Our task here is to solve for these transfer functions.

At this point, we make the following general notational definitions:

$$\underline{H}_k(s_1, s_2, \dots, s_k) = \begin{bmatrix} H_k^1(s_1, \dots, s_k) \\ H_k^2(s_1, \dots, s_k) \\ \vdots \\ H_k^m(s_1, s_2, \dots, s_k) \end{bmatrix} \quad (2-50)$$

where

$H_k^j \equiv$ k th order nonlinear transfer function from the input to the j th port;
 $m = 8$ in our example here.

$$\underline{v}(t) = [v_1(t) \ v_2(t) \ \dots \ v_m(t)]^T \quad (2-51)$$

where $v_i \equiv$ voltage at the i th port

The cutset equations for the m -port nonlinear network can be written as:

$$\underline{Y}(p)\underline{v} + \underline{F}(\underline{v}) + \underline{G}(\underline{u}, \underline{v}) + p\underline{Q}(\underline{v}) + \frac{1}{p}\underline{\phi}(\underline{v}) = [\underline{v}_g/z_g(p)][1 \ 0 \ 0 \ \dots \ 0]^T \quad (2-52)$$

where

$p \equiv$ differential operator, $\frac{d}{dt}$

$\underline{Y}(p) \equiv$ Reduced admittance matrix for the p -port augmented linear network

$\underline{F}(\underline{v}) \equiv$ vector composed of all nonlinear currents through the zero memory independent nonlinearity

$\underline{G}(\underline{u}, \underline{v}) \equiv$ vector composed of all nonlinear currents through the zero memory dependent nonlinearities

$\underline{Q}(\underline{v}) \equiv$ vector composed of all nonlinear currents through the nonlinear capacitive nonlinearities

$\underline{\phi}(\underline{v}) \equiv$ vector composed of all nonlinear currents through the nonlinear inductive elements.

$z_g(p) \equiv$ source impedance

Since the linear parts of the functions $F(\cdot)$, $G(\cdot)$, $Q(\cdot)$, and $\phi(\cdot)$ in eqn. (2-46) through (2-49) have been lumped together with the linear part of the network, the general form of these functions will be as follows:

$$\underline{Z}(\underline{v}) = \underline{Z}_2(\underline{v}) + \underline{Z}_3(\underline{v}) + \underline{Z}_4(\underline{v}) + \dots \quad (2-53)$$

where

$\underline{Z}_2(\underline{v})$ is a quadratic function of \underline{v}

$\underline{Z}_3(\underline{v})$ is a cubic function of \underline{v}

$\underline{Z}_4(\underline{v})$ is a quartic function of \underline{v}

...

$\underline{Z}(\cdot)$ being $\underline{F}(\cdot)$, $\underline{G}(\cdot)$, $\underline{Q}(\cdot)$, or $\underline{\phi}(\cdot)$. Thus, eqn. (2-53) can be re-written as:

$$\underline{Y}(p)\underline{v} = \frac{1}{\underline{z}_g(p)} \begin{bmatrix} v_g(t) \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} - \underline{i}_k(t) \quad , \quad k \geq 2 \quad (2-54)$$

where $\underline{i}_k(t)$ denotes vectors of 2nd and higher order current sources due to $\underline{F}(\underline{v})$, $\underline{G}(\underline{u}, \underline{v})$, $p\underline{Q}(\underline{v})$, and $\frac{1}{p}\underline{\phi}(\underline{v})$. The mathematical artifice used in section (2-2) could have been applied here also to obtain the form of all the non-linear current source terms, $\underline{i}_k(t)$. For the sake of brevity, we will not use that approach here, but simply use the results of section (2-2) to identify the different order current sources due to different nonlinearities. These are summarized in Table 2-1, where $v^i(t)$ denotes the i th order response voltage $v(t)$, which control the nonlinear element characteristics.

Table 2-1. Nonlinear Current Sources in multiple-node, multiple-nonlinearity circuit analysis.

Nonlinear Resistor, $F(v)$:

$$k = 2: a_2[v^1]^2$$

$$k = 3: 2a_2[v^1v^2] + a_3[v^1]^3$$

$$k = 4: a_2[2v^1v^3 + (v^2)^2] + 3a_3[v^1]^2v^2 + a_4[v^1]^4$$

Nonlinear Dependent Nonlinearity $G(u, v)$:

$$k = 2: a_{20}[u^1]^2 + a_{02}[v^1]^2 + a_{11}u^1v^1$$

$$k = 3: a_{30}[u^1]^3 + a_{03}[v^1]^3 + a_{21}[u^1]^2v^1 + a_{12}u^1[v^1]^2 + 2a_{20}u^1u^2 + 2a_{02}v^1v^2 + a_{11}[u^1v^2 + u^2v^1]$$

$$k = 4: a_{40}[u^1]^4 + a_{04}[v^1]^4 + a_{13}u^1[v^1]^3 + a_{22}[u^1]^2[v^1]^2 + a_{21}(2u^1u^3 + [u^2]^2) + a_{11}(u^3v^1 + u^1v^3 + u^2v^2) + a_{02}(2v^1v^3 + [v^2]^2) + 3a_{30}[u^1]^2v^2 + 3a_{03}[v^1]^2v^2 + a_{21}([u^1]^2v^2 + 2u^1u^2v^1) + a_{12}(u^2[v^1]^2 + 2u^1v^1v^2)$$

Nonlinear Capacitive Nonlinearity $pQ(v)$:

$$k = 2: a_{2p}[v^1]^2$$

$$k = 3: 2a_{2p}[v^1v^2] + a_{3p}[v^1]^3$$

$$k = 4: a_{2p}(2v^1v^3 + [v^2]^2) + 3a_{3p}[v^1]^2v^2 + a_{4p}[v^1]^4$$

Table 2-1 (contd.)

Nonlinear Inductive Nonlinearity, $[1/p]\phi(v)$

$$k = 2: \frac{a_2}{p} [v^1]^2$$

$$k = 3: \frac{2a_2}{p} [v^1 v^2] + \frac{a_3}{p} [v^1]^3$$

$$k = 4: \frac{a_2}{p} (2v^1 v^3 + [v^2]^2) + \frac{3a_3}{p} [v^1]^2 v^2 + \frac{a_4}{p} [v^1]^4$$

We observe that the nonlinear current source terms in Table 2-1 are similar to the nonlinear terms whose transforms were derived in section 2-2, except for the nonlinear dependent source terms, which are functions of two controlling voltages u and v . The form of the transforms of the nonlinear dependent source will, however, be similar to the other nonlinearity types. These can again be written by inspection. For example,

$$a_{20}[u^1(t)]^2 + a_{20}u^1(t_1)u^1(t_2) \leftrightarrow a_{20}U(s_1)U(s_2) \quad (2-55)$$

$$a_{11}u^1(t)v^1(t) + a_{11}u^1(t_1)v^1(t_2) \leftrightarrow a_{11}\overline{U(s_1)V(s_2)} \quad (2-56)$$

$$a_{20}u^1(t)v^2(t) + a_{20}u^1(t_1)v^2(t_2, t_3) \leftrightarrow a_{20}\overline{U(s_1)V(s_2, s_3)} \quad (2-57)$$

Recall the one way arrow goes backwards only when $t_1=t_2=t_3$.

We also note that a k -th order current source term in Table 2-1 depends on responses of order less than k , which implies that, in order to calculate a transfer function of order k , we need to determine the transfer function up to order $(k-1)$.

The first order transfer function can be solved for easily. It is simply the linear circuit response. Therefore,

$$\underline{Y}(p)\underline{v}(t) = \underline{i}_1(t) \quad (2-58)$$

For a single input system, $\underline{i}_1(t) = 1/z_g [v_g(t) \ 0 \ 0 \ \dots \ 0]^T$, where $v_g(t)$ is the source voltage. Taking the transform of eqn. (2-58), and assuming that the input source to be an impulse function, we get:

$$\underline{v}^1(s_1) = \underline{H}_1(s_1) = 1/z_g [\underline{Y}(s_1)]^{-1} [1 \ 0 \ 0 \ \dots \ 0]^T \quad (2-59)$$

where $\underline{H}_1(s_1)$ was defined in eqn. (2-50).

The equation for obtaining the second-order response, as per eqn. (2-54), is the following:

$$\underline{Y}(p)\underline{V}^{(2)}(t) = -\underline{i}_2(t) \quad (2-60)$$

Since the input to the nonlinear circuit is assumed to be an impulse function, the transform of eqn. (2-60), after using eqn. (2-14), is:

$$\underline{Y}(s_1+s_2)\underline{H}_2(s_1,s_2) = -\underline{I}_2(s_1,s_2) \quad (2-61)$$

The elements of vector $\underline{I}_2(s_1,s_2)$ can be obtained by performing a two-dimensional transform on the terms associated with $k = 2$ in Table 2-1. This operation, as indicated earlier, can be carried out by inspection. Thus, we have

$$\underline{H}_2(s_1,s_2) = -[\underline{Y}(s_1+s_2)]^{-1}\underline{I}_2(s_1,s_2) \quad (2-62)$$

Likewise we can solve for $\underline{H}_3(s_1,s_2,s_3)$. In general, we solve for the n th order transfer function using eqn. (2-63):

$$\underline{H}_n(s_1,s_2,\dots,s_n) = [\underline{Y}(\sum_{i=1}^n s_i)]^{-1}\underline{I}_n(s_1,\dots,s_n) \quad (2-63)$$

We observe a striking similarity between eqn. (2-63) and the equations for nodal or cutset analysis encountered in linear circuit analysis. A little thought would show that the process of solving eqn. (2-63) is identical to solving the linear circuit in Fig. 2-2. We have nonlinear current sources as inputs to the augmented linear circuit. A k -th order vector of transfer functions is obtained by exciting the linear circuit by the k th order current sources. Just as in the case of linear systems, superposition can be applied here when a particular order response is determined from the

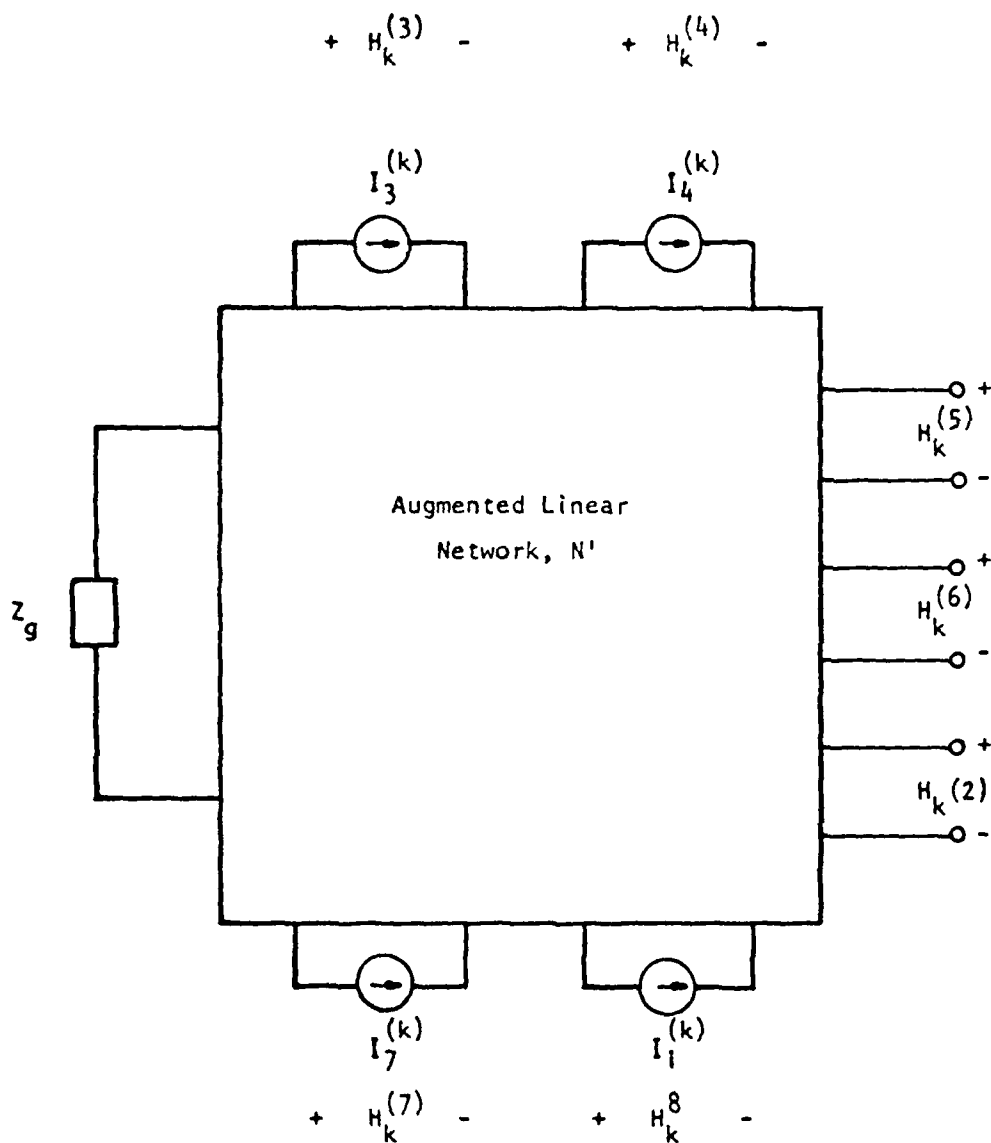


Figure 2-2. Determination of Volterra Transfer Functions.

lower order responses. That is, a k-th order response can be obtained by applying the k-th order current sources one-by-one at each of the ports and then summing up the responses. It is important to note, however, that the complete responses of order up to (k-1) must be determined before we can obtain the kth order response by superposition. It is also noted that the illustration of Fig. 2-2 is for pedagogic purpose and that the nonlinear current sources are not physically present in the circuit under consideration.

2-5. Multiple Input Circuit Analysis

Much of the foregoing discussion has been concerned with the analysis of nonlinear circuits with single inputs. However, many applications of practical significance in nonlinear circuit analysis have multiple inputs. For example, in a receiver system, the mixer circuit has two inputs: 1) the message signal, and 2) the local oscillator signal. The transmitter again has nonlinear circuits with multiple inputs. The Volterra series method is especially well suited for the analysis of such circuits. In this section we discuss how the various order transfer functions change as a result of multiple inputs.

From the discussion in section 2-4, it should be apparent that the analysis of nonlinear circuits using the Volterra series method involves the repeated analysis of a linearized circuit. The fundamental relationship had the following form (see eqn. 2-54):

$$\underline{Y}(p)\underline{V}(t) = \frac{1}{Z_g(p)} \underline{i}_1(t) - \underline{i}_2(t) - \underline{i}_3(t) + \dots \quad (2-64)$$

where $\underline{i}_k(t)$ is the k-th current source vector. For $k \geq 2$ the k-th order current source, depends on up to the (k-1) order voltage ratio transfer

functions as discussed above. It is injected at each of the pots at which the nonlinear elements are present, and is due entirely to the nonlinear characteristics of the nonlinearity. Furthermore, it is proportional to the k values of the circuit input multiplied together. Thus, the number of elements in the vector $\underline{i}_k(t)$, $k \geq 2$, remain unchanged when multiple inputs are present; only the $\underline{i}_1(t)$ vector is changed.

Consider, for example, the two-input circuit of Fig. 2-3(a). Then, to solve for the first-order transfer function, we write the vector transform equation as:

$$\underline{Y}(s_1)\underline{V}(s_1) = \underline{I}_1(s_1) \quad (2-65)$$

where

$$\underline{I}_1(s_1) = [Y_{g1}(s_1)V_{g1}(s_1) \quad Y_{g2}(s_1)V_{g2}(s_1) \quad 0 \quad \dots \quad 0]^T \quad (2-66)$$

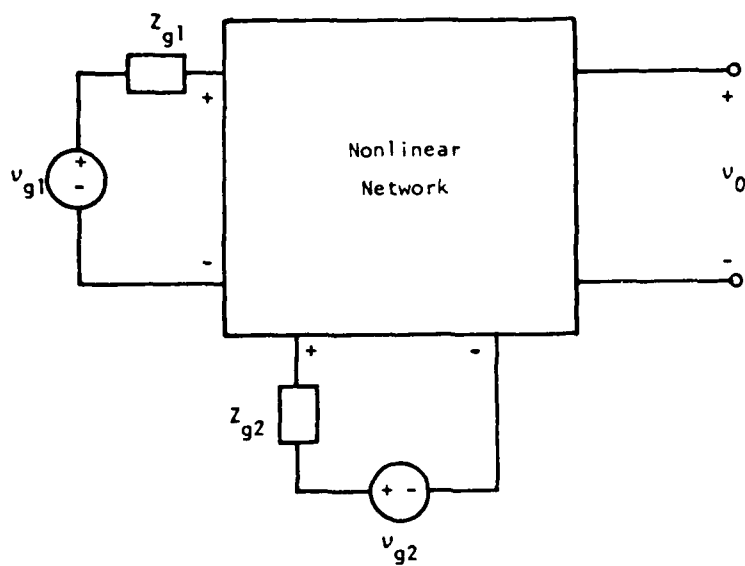
and \underline{Y} and \underline{V} are as defined previously. The transfer function vector can be written as:

$$\underline{H}_1(s_1) = \underline{H}_{10}(s_1) + H_{01}(s_1) \quad (2-67)$$

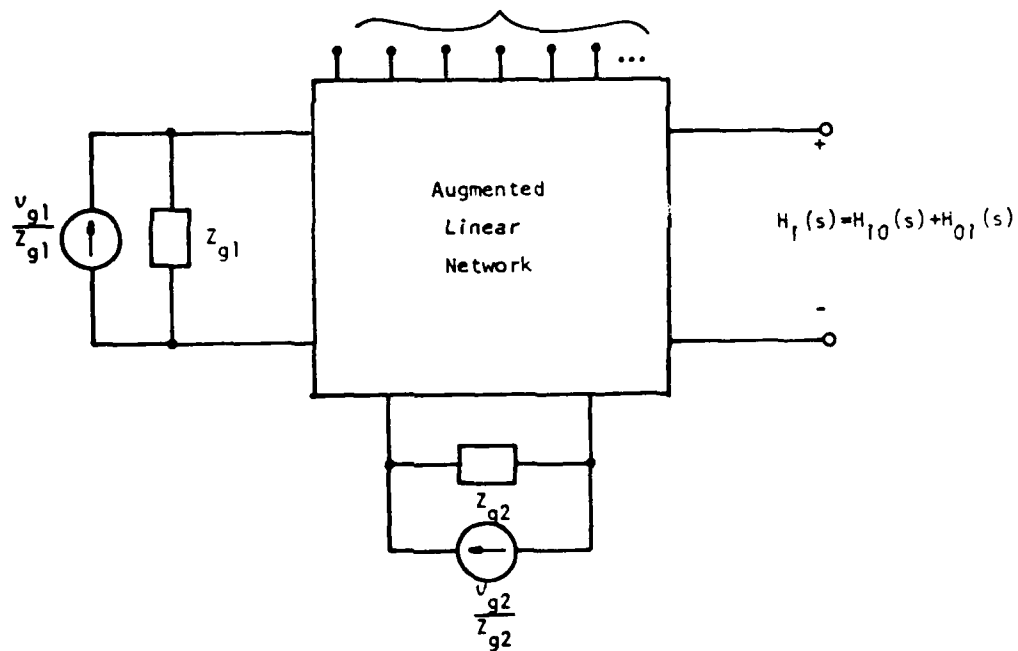
where

$$\underline{H}_{10}(s_1) = \left[\frac{V^{(1)}(s_1)}{V_{g1}} \quad \frac{V^{(2)}(s_1)}{V_{g1}} \quad \dots \quad \frac{V^p(s_1)}{V_{g1}} \right]^T \bigg|_{V_{g2}=0} \quad (2-68)$$

and

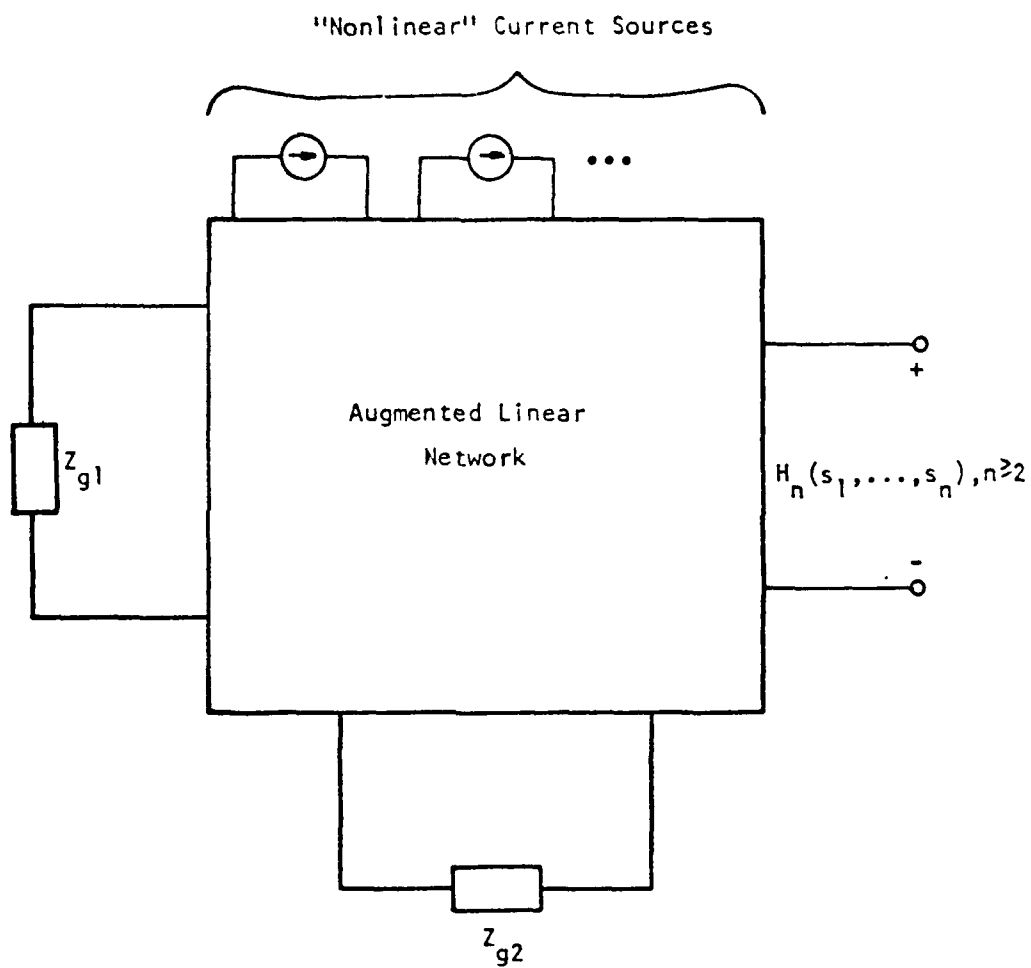


Nonlinear element controlling voltage ports.



(b) Circuit for determining first-order transfer function.

Fig. 2-3. Multiple Input Nonlinear Circuit Analysis.



(c) Circuit for determining $H_n(s_1, s_2, \dots, s_n), n \geq 2$.

Fig. 2-3. (contd.) Multiple Input Nonlinear Circuit Analysis.

$$\underline{H}_{01}(s_1) = \left[\frac{V^{(1)}(s_1)}{V_{g2}} \quad \frac{V^{(2)}(s_1)}{V_{g2}} \quad \dots \quad \frac{V^{(p)}(s_1)}{V_{g1}} \right] T \bigg|_{V_{g1}=0} \quad (2-69)$$

where $V^{(i)}$ is the voltage at port i .

The second- and higher-order transfer function vectors are solved for by removing the given input sources and applying the fictitious nonlinear current sources across the ports at which the nonlinear elements are present. The vector transform equation for solving for the second-order transfer function is still given by:

$$\underline{H}_2(s_1, s_2) = -[\underline{Y}(s_1 + s_2)]^{-1} [\underline{I}_2(s_1, s_2)] \quad (2-70)$$

where

$$\underline{I}_2(s_1, s_2) = [I^{(1)}(s_1, s_2) \quad I^{(2)}(s_1, s_2) \quad \dots \quad I^{(p)}(s_1, s_2)] \quad (2-71)$$

Depending on the nonlinearity type, the general form of $I^{(l)}(s_1, s_2)$, the second-order current source across port l , will be:

$$I_2^{(l)}(s_1, s_2) = a_2 H_1^{(l)}(s_1) H_1^{(l)}(s_2) \quad (2-72)$$

where $H_1^{(l)}(\cdot)$ is known from eqn. (2-67). The determination of the higher-order transfer functions is done similarly.

In summary, we note that the presence of multiple input sources in a nonlinear circuit does not drastically alter the procedure for determining the Volterra transfer functions. Only the structure of the first-order current source vector is changed as a result of multiple sources. This change is reflected in the values of the elements making up the second- and higher-order current source vectors, whose structure remains unchanged.

2-6. Sinusoidal Steady-State Analysis

In linear system theory, the sinusoidal steady-state response is intimately tied to the transfer function of the system. A similar result is found for higher order responses using the Volterra series method: an n -th order response at a particular frequency is directly related to the n -th order transfer function. In this section we develop this relationship.

If the harmonic input method [10-12] had been used in deriving the generalized transfer functions in the previous sections, the relationship between the n -th order steady state response and the n -th order transfer function would have been self-evident. But, since multi-dimensional transform theory was used to derive the generalized transfer functions, this relationship must be developed. We treat the specific case of $n=2$ in section 2-6.1 and then derive the general relationship in section 2-6.2.

2-6.1. Second-order Sinusoidal response:

The second-order output, according to the Volterra series, is given by:

$$y_2(t) = \int_0^\infty \int_0^\infty h_2(t-\tau_1, t-\tau_2) x(\tau_1) x(\tau_2) d\tau_1 d\tau_2 \quad (2-73)$$

Consider the input signal comprising two unit sinusoidal signals at frequencies ω_a and ω_b . The input $x(\tau)$ is therefore:

$$x(\tau) = \left[\frac{\exp(j\omega_a \tau) + \exp(-j\omega_a \tau)}{2} \right] + \left[\frac{\exp(j\omega_b \tau) + \exp(-j\omega_b \tau)}{2} \right] \quad (2-74)$$

Substituting eqn. (2-74) in (2-73), we have:

$$\begin{aligned}
y_2(t) = & \int_0^\infty \int_0^\infty h_2(t-\tau_1, t-\tau_2) \cdot \\
& \cdot \left[\frac{\exp(j\omega_a \tau_1) + \exp(-j\omega_a \tau_1)}{2} + \frac{\exp(j\omega_b \tau_1) + \exp(-j\omega_b \tau_1)}{2} \right] \\
& \cdot \left[\frac{\exp(j\omega_a \tau_2) + \exp(-j\omega_a \tau_2)}{2} + \frac{\exp(j\omega_b \tau_2) + \exp(-j\omega_b \tau_2)}{2} \right] \\
& \cdot d\tau_1 d\tau_2
\end{aligned} \tag{2-75}$$

Considering one cross term only,

$$\int_0^\infty \int_0^\infty h_2(t-\tau_1, t-\tau_2) \frac{1}{4} \exp(j\omega_a \tau_1 + j\omega_b \tau_2) d\tau_1 d\tau_2 \tag{2-76}$$

and letting $\sigma_1 = t - \tau_1$ and $\sigma_2 = t - \tau_2$ and carrying out the integration yields,

$$\frac{1}{4} H_2(j\omega_a, j\omega_b) \exp[j(\omega_a + \omega_b)t] \tag{2-77}$$

Considering the other cross term similarly yields

$$\frac{1}{4} H_2(j\omega_b, j\omega_a) \exp[j(\omega_a + \omega_b)t] \tag{2-78}$$

However, if $H_2(s_1, s_2)$ is symmetrical in its arguments, as they are assumed to be in this report, then the terms in eqns. (2-77) and (2-78) are equal. The complex conjugate terms appear similarly. Hence, the output at frequency $\omega_a + \omega_b$ is:

$$y(t)|_{\omega_a + \omega_b} = |H_2(j\omega_a, j\omega_b)| \cos[(\omega_a + \omega_b)t + \theta_{a+b}] \tag{2-79}$$

The $2\omega_a$ or $2\omega_b$ term and their complex conjugates appear only once in eqn. (2-75); hence, their magnitude will be $\frac{1}{2}|H_2(j\omega_a, j\omega_a)|$ and $\frac{1}{2}|H_2(j\omega_b, j\omega_b)|$, respectively. If only one frequency input was present, the results would be similar. The second-order output would then be:

$$y_2(t) = |H_2(j\omega_a, -j\omega_a)| + \frac{|H_2(j\omega_a, j\omega_a)|}{2} \cos(2\omega_a t + \theta_{2a}) \quad (2-80)$$

Thus, if we know $H_2(s_1, s_2)$, then the quantities in eqn. (2-80) can be easily evaluated. This is analogous to the case of linear systems, where the complex variable s is replaced by $j\omega$ to compute the response at ω .

If more than two-tones were present at the input, the second order response would be evaluated by taking all combinations of two frequencies at a time.

The response of the third and higher orders is similarly treated. We now present the general case.

2-6.2. General Sinusoidal Steady-State Analysis.

In this sub-section, we develop the relationship which can be applied directly to compute the sinusoidal steady-state response of a nonlinear system from its nonlinear transfer functions, which can be obtained by the method presented in section 2. The discussion here relies heavily on [10].

Consider a nonlinear system excited by the sum of K distinct tones; i.e., defining $N = 2K$, we have,

$$x(t) = \frac{1}{2} \sum_{i=1}^N A_i \exp(j\omega_i t) \quad (2-81)$$

where ω_i will include both positive and negative frequencies, and A_i for a negative frequency will be the complex conjugate of A_i for the positive frequency.

quency in order to have $x(t)$ real. Then, the n th order output, $y_n(t)$, is given by:

$$y_n(t) = \int \cdots \int_{n\text{-fold}} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i) d\tau_i$$

$$= \int \cdots \int h_n(\tau_1, \dots, \tau_n) \frac{1}{2^n} \sum_{i=1}^n \sum_{k=1}^N A_k \exp[j\omega_k(t-\tau_i)] d\tau_i \quad (2-82)$$

Carrying out the product operation in eqn. (2-82), we get a function $y_n(t)$ containing N^n terms, given by:

$$y_n(t) = \sum_{k_1=1}^N \cdots \sum_{k_n=1}^N \frac{1}{2^n} A_{k_1} \cdots A_{k_n} H_n(j\omega_{k_1}, \dots, j\omega_{k_n})$$

$$\cdot \exp[j(\omega_{k_1} + \dots + \omega_{k_n})t] \quad (2-83)$$

Notice that in arriving at eqn. (2-83), we have performed the τ_i integration in eqn. (2-82), thus giving rise to the n -th order transfer function in eqn. (2-83). As the indices k_i are varied over the range 1 to N , many of the terms will be at the same frequency. The number of terms at various particular frequencies will vary according to what frequency combinations are taken. For example, in the case of $n=2$ in section 2-6.1, there were two cross frequency terms, while there was only one second harmonic (at $2\omega_a$) term. Similarly, for $n=3$, there are six terms in eqn. (2-83) at frequency $\omega_a + \omega_b + \omega_c$, three terms at $2\omega_a + \omega_b$, one term at $3\omega_a$, etc. The nonlinear transfer functions, which make up the coefficients of these frequency terms, differ only in their arguments. However, since the transfer functions are assumed to be symmetric, the coefficient of the output at frequency $\omega_a + \omega_b + \omega_c$

(in the case of $n=3$) can be multiplied by 6. This obviates the need for taking all combinations to compute the output at $\omega_a + \omega_b + \omega_c$. Likewise we handle the case of other frequency combinations. With this insight, we can peek at the problem from a different perspective.

Let m_1, m_2, \dots, m_N be non-negative integers. Then, the number of terms at frequency $\omega_\Sigma = m_1\omega_1 + m_2\omega_2 + \dots + m_N\omega_N$ is equal to the number of ways of forming $m_1\omega_1 + \dots + m_N\omega_N$. In the n -th order output spectrum to a multi-tone input, each term is evaluated by taking a distinct combination of n input tones at a time. To compute the n -th order output when the input frequencies are $\omega_1, \omega_2, \dots, \omega_N$, we must therefore restrict m_i in the following manner to compute ω_Σ :

$$m_1 + m_2 + \dots + m_N = n \quad (2-84)$$

Now the problem reduces to the following: find the number of ways in which n objects can be divided into N groups of which the first contains m_1 objects, the second m_2 objects, etc. The solution to this problem is given by the multi-nomial coefficient [22]:

$$C_{n,N} = \frac{n!}{m_1! m_2! \dots m_N!} \quad (2-85)$$

By deriving eqn. (2-85), we have obviated the repetition of terms that is inherent in eqn. (2-83). An equivalent way of representing eqn. (2-83) through the use of eqn. (2-85) then becomes:

$$y_n(t) = \sum_{n,N} C_{n,N} \frac{A_1^{m_1} A_2^{m_2} \dots A_N^{m_N}}{2^n}$$

$$\cdot H_n(j\omega_1, \dots, j\omega_2, \dots, j\omega_N, \dots) \\ m_1 \text{ times } m_2 \text{ times } m_N \text{ times}$$

$$\cdot \exp[j(m_1\omega_1 + \dots + m_N\omega_N)t] \quad (2-86)$$

Since $y_n(t)$ is real, eqn. (2-86) also contains the complex conjugate terms. Thus, the coefficient of the sinusoidal term at frequency $m_1\omega_1 + \dots + m_N\omega_N$ in the n -th order output is given by:

$$c_{n,N} \frac{A_1^{m_1} A_2^{m_2} \dots A_N^{m_N}}{2^{n-1}} H_n(j\omega_1, \dots, j\omega_2, \dots, j\omega_N, \dots) \quad (2-87) \\ m_1 \text{ times } m_2 \text{ times } m_N \text{ times}$$

In computing the entire n -th order response in eqn. (2-86), we take all distinguishable combinations of m_i satisfying eq. 82-84). According to [10] there are

$$S_{n,N} = \binom{n+N-1}{n} = \frac{(n+N-1)!}{n!(N-1)!} \quad (2-88)$$

such combinations.

Equation (2-86) is the fundamental relationship between the n -th order output and the n -th order transfer function. At first glance, the evaluation of this equation appears to be a formidable task. But, after some thought, one finds that this is not such a difficult task after all. We, however, defer the discussion of this till section 4.

We now illustrate the use of eqn. (2-87). We assume that the nonlinear transfer functions are known. The case for $n=2$ can be easily verified from the discussion in section 2-6.1. For a two-tone input at ω_1 and ω_2 and $n=3$, we have the following cases:

(a) The output at ω_1 and ω_2 have the following amplitudes, respectively:

$$y_3(t)|_{\omega_1} = \frac{3!|A_2|^2 A_1}{(4)1!1!1!} |H_3(j\omega_1, -j\omega_2, j\omega_2)| \quad (2-89)$$

$$y_3(t)|_{\omega_2} = \frac{3!A_2|A_1|^2}{(4)1!1!1!} |H_3(j\omega_1, -j\omega_1, j\omega_2)| \quad (2-90)$$

(b) The output at $2\omega_1 + \omega_2$ has the following magnitude:

$$y_3(t)|_{2\omega_1 + \omega_2} = \frac{3!A_1^2 A_2}{(4)2!1!} |H_3(j\omega_1, j\omega_1, j\omega_2)| \quad (2-91)$$

(c) The output at $3\omega_1$ has the following magnitude:

$$y_3(t)|_{3\omega_1} = \frac{3!(A_1)^3}{(4)3!} |H_3(j\omega_1, j\omega_1, j\omega_1)| \quad (2-92)$$

The other combinations can be carried out similarly. For the above cases we make the following observations: both eqns. (2-89) and (2-90) are similar to obtaining the output at $\omega_a + \omega_b + \omega_c$, and therefore we see a $3!$ ($=6$) multiplication factor*, which accounts for the six combinations at $\omega_a + \omega_b + \omega_c$ that were mentioned earlier; eqn. (2-91) is similar to obtaining the output at $2\omega_a + \omega_b$, and therefore has a multiplication factor of $(3!/2!) = 3$, which again is in accordance with our earlier discussion; eqn. (2-92) is like evaluating the output at $3\omega_a$, and hence has a multiplication factor of $(3!/3!) = 1$.

In section (2-5), we dealt with the analysis of multiple input non-linear circuits. In obtaining the sinusoidal steady-state response of such

*The constant factor 4 in the denominator appears consistently in all the output terms, and is therefore not regarded as a variable multiplication factor here. This factor appears due to the way $x(t)$ was expressed in eqn. (2-81).

circuits the material of this section is still applicable. However, care must be taken in keeping track of the various input frequencies, and their associated transfer functions, when such an analysis is warranted.

CHAPTER 3

COMPUTER-AIDED ANALYSIS USING VOLTERRA SERIES

3-1. Introduction

The adapting of Volterra series method in a general simulation program has been regarded as difficult by various authors [30]. As such, virtually no effort has been spent on investigating the computational aspect of this method. Most previous works, such as [7], have endeavored to check the validity of this approach by applying it to specific circuit problems using a computer.

The only major effort in using the Volterra series for general nonlinear circuit analysis has been the development of the program NCAP [10,24]. A cursory review of this program reveals the inherent inefficiency in the computational approach with regards to storage and types of algorithms used. This inefficiency notwithstanding, there are severe limitations regarding the usefulness of the program: first, the program merely computes the numerical values of the nonlinear transfer function at the various program-prescribed combinations of the input frequencies, and does not compute all the transfer function values which are required to compute the complete output spectrum. Thus, NCAP does not yield the entire output spectrum information. Second, to compute up to an n -th order transfer function, the user must specify n input frequencies, which are assumed to be a sum of exponentials and not real sinusoids. The program, therefore, is severely limited in its usefulness from the point of view of a user who may only be interested in obtaining the output spectrum - say, for example, up to the third order response to two sinusoidal inputs - and has little use for the

numerical values of the transfer functions at the program prescribed frequencies.

In this section we look at the computational aspect of the Volterra series method for general simulation purposes and then present the basic algorithms for adapting this method for the spectrum and distortion analysis of nonlinear circuits with polynomial type nonlinearities.

In section 3-2, we present a brief overview of symbolic analysis in linear circuits, and then describe the reason why a symbolic approach is particularly useful in adapting Volterra series for general simulation. Section 3-3 deals with the implementation of the symbolic approach, and also contrasts the computational effort between a numerical approach and the particular symbolic approach used here. The algorithm for obtaining the complete output spectrum and the various distortion indices is described in section 3-4. A description of the computer implementation of these algorithms is given in section 3-5.

3-2. Why a Symbolic Analysis Approach.

The symbolic analysis of circuits involves the computation of the a_i and b_i for network functions in the form

$$F(s) = \frac{N(s)}{D(s)} = \frac{\sum a_i s^i}{\sum b_i s^i} \quad (3-1)$$

when all circuit elements are known. The more general form

$$F(s; x_1, x_2, \dots, x_n) = \frac{N(s; x_1, \dots, x_n)}{D(s; x_1, \dots, x_n)} \quad (3-2)$$

applies when some elements of the circuit x_i are kept as symbols. The advantages of symbolic analysis have been recognized previously [25,27]. One

particular advantage, and the one which is relevant to our problem here, is that the numerical evaluation of a function at discrete points is much easier and faster once the symbolic function is obtained than working repeatedly with a circuit analysis program. With this brief overview of symbolic analysis, we now proceed to answer the question: Why use a symbolic analysis approach for adapting the Volterra series method for general circuit analysis?

As pointed out in the previous sections, a nonlinear circuit is completely characterized by its Volterra kernels, or their transforms - the generalized transfer functions. These transfer functions are then directly related to the various order sinusoidal steady-state responses, as described in Chapter 2. The n -th order transfer function is determined from the following equation (see Chapter 2):

$$H_n(s_1, \dots, s_n) = [Y(\sum_{i=1}^n s_i)]^{-1} I_n(s_1, \dots, s_n) \quad (3-3)$$

where $Y(\sum_{i=1}^n s_i)$ is the reduced node admittance matrix evaluated as $s_1 + s_2 + \dots + s_n$, and I_n is the n -th order current source vector due to the nonlinear elements. To compute the output spectrum, we evaluate H_n at the various and many frequency combinations. From eqn. (3-3) it should be clear that such an evaluation will entail the inversion of the reduced node admittance matrix at each of these frequency combinations. Using combinatorial analysis, it has been shown [22] that for an input consisting of M sine waves, the number of inversions involved in an n -th order response, given by $N_{n,m}$, is:

$$N_{n,m} = \binom{2M+n-1}{n} \quad (3-4)$$

Thus, for a 3-tone input and up to a third order analysis, the number of inversions is approximately 285. For higher order responses, this number grows very rapidly.

Two basic approaches available for handling this inversion process are: 1. Numerical approach, or 2. Symbolic approach. The advantage of evaluating symbolic transfer functions mentioned earlier makes the symbolic approach more attractive. How much advantage is gained in using a symbolic analysis depends on how much computational effort is expended in obtaining the symbolic inverse of the reduced node admittance matrix. Thus, an efficient scheme for obtaining the symbolic inverse must be used to efficiently adapt the Volterra series method for computer aided analysis. The determination of the symbolic inverse will be the subject of section 3-3.

The reasons presented above stem from looking at the computational aspect of adapting Volterra series for computer-aided analysis. There are other advantages gained from using a symbolic analysis. An important one is that the generalized transfer functions can be obtained as functions of s_i once the inverse of the reduced node admittance matrix is obtained as a symbolic function of s . This can be seen from examining eqn. (3-3). The formation of the n -th order current source vector is a bootstrapping operation, as was pointed out in Chapter 2. That is, an n -th order source is formed from transfer functions of order less than n . The first-order transfer function vector is determined from a column^{*} of the symbolic inverse of the reduced node admittance matrix. The second order current sources, which

*This is assuming a single input circuit.

depend on the elements of the first order transfer function vector, are therefore formed from this column of $[Y(s)]^{-1}$. The second-order transfer function vector is obtained by pre-multiplying the second-order current source vector by $[Y(s_1+s_2)]^{-1}$, according to which the second-order transfer function vector eventually depends on the entries of inverse of the node admittance matrix evaluated at (s_1+s_2) . The third- and higher-order transfer functions have a similar dependence. Thus, an inverse of the reduced node admittance matrix in symbolic form, with s retained as a symbol, also yields a functional description of the nonlinear transfer functions. A concomitant advantage of this functional description is that theorems from multi-dimensional theory [5] (such as initial value, final value, etc.) can then be used to gain more insight into the workings of the circuit.

[23] has developed recursive relationships to estimate the error incurred in the truncation of the series solution. This error was directly related to the l_1 norm of the linear kernel function, which, in turn, is related to the poles and residues of the linearized system. Thus, we can get an estimate of the accuracy of our solution through the pole-residue information provided to us by the symbolic analysis.

3-3. Symbolic Analysis Method

Symbolic circuit analysis by digital computer has been of considerable interest in the past decade. Many algorithms and methods have been derived to obtain symbolic transfer functions of linear circuits [20]. Most of these methods use tree enumeration [26], signal-flow graphs [20], or purely numerical methods [27] to obtain symbolic transfer function between the input and the output. These approaches are basically useful for single-input, single-output systems. The inversion of the reduced node admittance matrix to obtain the open-circuit impedance matrix, which is the problem we are

dealing with, is basically a multi-input, multi-output problem. The methods mentioned above can be adapted for solving the problem at hand; however, the generation of multiple symbolic functions using these approaches may not be satisfactory because of excessive computer time requirements. Some other approach is definitely warranted.

Published methods [16-18] for inverting the nodal admittance matrix when the elements are rational functions of the Laplace transform variable s use pivotal techniques. It may appear that, since it is easy to program a computer to perform polynomial arithmetic, these pivotal-techniques are a natural way to approach the symbolic inversion problem. Results from the use of such a technique have proved to be disappointing, mainly due to the following reasons:

(a) The process of inversion transforms the nodal admittance matrix, which contains terms of the form $a + \frac{b}{s} + c$, into a matrix in which every element is a rational function of s . The pivotal technique produces the inverse matrix where common factors appear between numerator and denominator, and unless some mechanism is built into the process whereby these common factors are recognized and removed, the elements produced will have polynomials of excessively high order.

(b) When the circuit complexity is high, the evaluation of the symbolic function at high frequency values can give rise to numerical problems. For example, a circuit with 8 poles will have an s^8 term in the characteristic polynomial. When evaluated at 10 Mrad/sec, this term produces a number equal to 10^{56} . Of course, this problem can be alleviated by obtaining a partial fraction expansion (PFE) form for the transfer functions. But this again entails additional computations - not to mention the numerical instability problems involved in root finding.

(c) It has also been found that pivotal techniques become numerically unstable for higher order circuits.

We therefore seek another alternative for obtaining the symbolic form of the open circuit impedance matrix.

An approach based on the state variable formulation can be used to achieve this goal. Specifically, consider the general p-port augmented linear circuit of Fig. 3-1(a). We wish to solve for the transfer impedances, $z_{ij}(s)$, $i, j = 1, 2, \dots, p$, from the j-th port to the i-th port. Knowing these transfer impedances, we can write for the p-port:

$$\underline{V}(s) = \underline{Z}(s)\underline{I}(s) = [\underline{Y}(s)]^{-1} \underline{I}(s) \quad (3-5)$$

$$\text{where } \underline{V}(s) = [V_1(s) \ V_2(s) \ \dots \ V_p(s)] \quad (3-6)$$

$$\underline{Z}(s) = [z_{ij}(s)] \quad (3-7)$$

$$\text{and } \underline{I}(s) = [I_1(s) \ I_2(s) \ \dots \ I_p(s)] \quad (3-8)$$

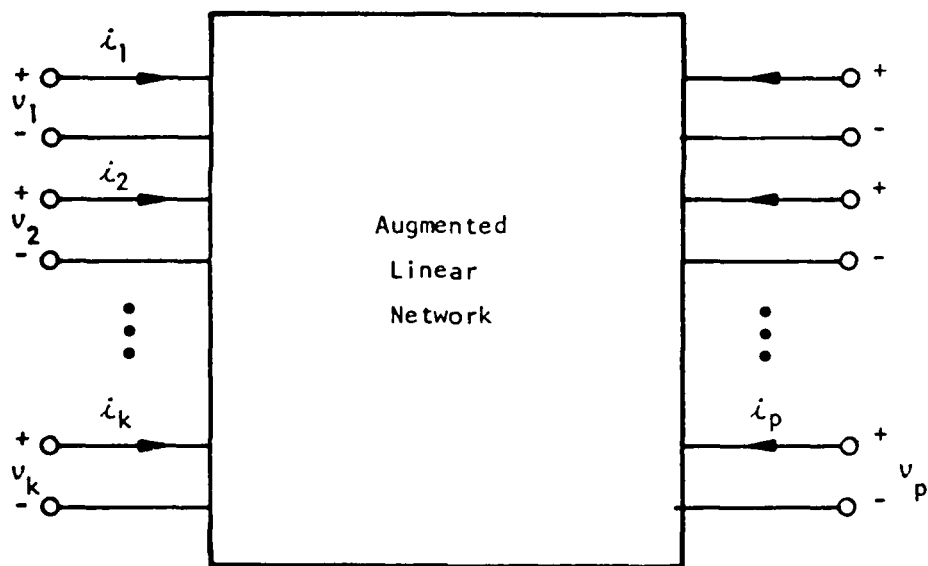
Note that the vector $\underline{V}(s)$ contains entries which are the output voltages and voltages that control the nonlinear element characteristics in the nonlinear circuit.

To obtain $\underline{Z}(s)$ symbolically, we write for the network of Fig. 3-1(b), the following state equations:

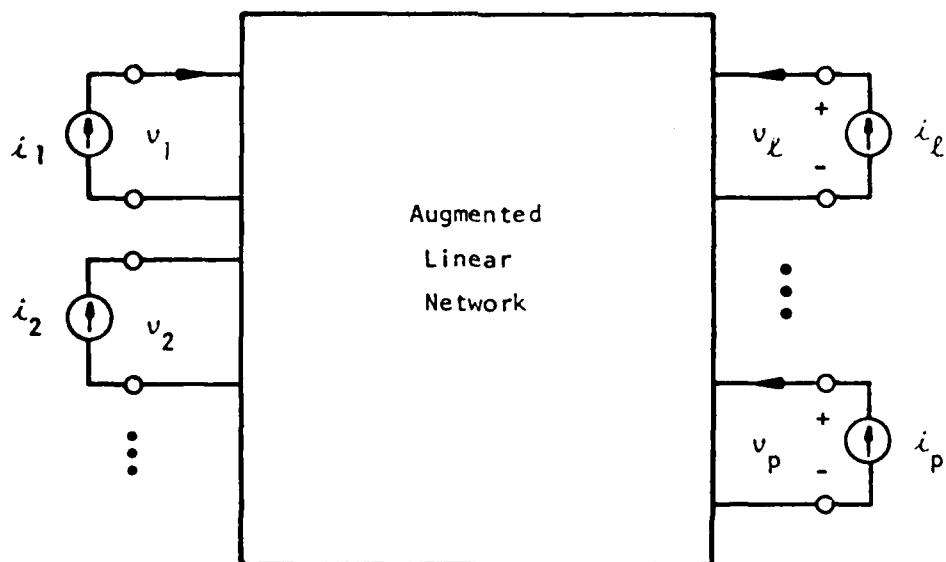
$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{i} \quad (3-9)$$

$$\underline{v} = \underline{C}\underline{x} + \underline{D}\underline{i} \quad (3-10)$$

where \underline{x} is the vector of state variables, and \underline{v} and \underline{i} are vectors whose



(a)



(b)

Figure 3-1. Determination of $[Y(s)]^{-1} = Z(s)$ for the p-port network using state equations.

transforms appear in eqns. (3-6) and (3-8), respectively. Taking the Laplace transform of eqn. (3-9) and (3-10), and solving for $V(s)$, we get:

$$\underline{V}(s) = [\underline{C}(s\underline{I}-\underline{A})^{-1} \underline{B} + \underline{D}] \underline{I}(s) \quad (3-11)$$

and, therefore, we get $Z(s)$ to be

$$\underline{Z}(s) \equiv [\underline{C}(s\underline{I}-\underline{A})^{-1} \underline{B} + \underline{D}] \quad (3-12)$$

which is identically the inverse of the reduced node admittance matrix.

The matrix $(s\underline{I}-\underline{A})$ can be inverted by applying the similarity transformation as follows:

$$\underline{A} = \underline{M} \underline{\Lambda} \underline{M}^{-1} \text{ or } \underline{\Lambda} = \underline{M}^{-1} \underline{A} \underline{M}$$

$$\therefore \underline{M}^{-1}(s\underline{I}-\underline{A})\underline{M} = s\underline{I} - \underline{M}^{-1}\underline{A}\underline{M} = s\underline{I} - \underline{\Lambda}$$

$$\text{or } (s\underline{I}-\underline{A})^{-1} = \underline{M}(s\underline{I}-\underline{\Lambda})^{-1} \underline{M}^{-1} \quad (3-13)$$

where the inverse of $(s\underline{I} - \underline{\Lambda})$ is simply $\text{diag} \{(s-\lambda_1)^{-1}, (s-\lambda_2)^{-1}, \dots\}$ where λ_i are the eigenvalues* of the \underline{A} matrix and \underline{M} is the modal matrix. Substituting eqn. (3-13) into eqn. (3-12), we get,

$$\begin{aligned} \underline{Z}(s) &= [\underline{C}\underline{M}(s\underline{I}-\underline{\Lambda})^{-1} \underline{M}^{-1} \underline{B} + \underline{D}] \\ &= [\hat{\underline{C}}(s\underline{I}-\underline{\Lambda})^{-1} \hat{\underline{B}} + \underline{D}] \end{aligned} \quad (3-14)$$

where $\hat{\underline{C}} \triangleq \underline{C}\underline{M}$ and $\hat{\underline{B}} \triangleq \underline{M}^{-1}\underline{B}$. Equation (3-14) yields the entries of $\underline{Z}(s)$ in partial fraction expansion form, which, as mentioned previously, is a more desirable form from a computational standpoint. All information about $\underline{Z}(s)$

*Here we assume distinct eigenvalues; the repeated eigenvalues can be handled similarly.

is contained in the matrices \hat{C} , \hat{B} , D and a vector containing the eigenvalues. An algorithm for implementing this approach is given in Fig. 3-2. It should be noted that the approach used here is completely numerical and does not involve any coding and decoding of symbols.

Now that an algorithm for obtaining the symbolic $Z(s)$ is defined, we can make a comparison of the computational effort involved between using a symbolic inverse and the numerical inverse of the node admittance matrix at each frequency point.

The computational trade-off between the symbolic approach and a numerical approach for matrix inversion is very problem dependent. While a clear-cut winner cannot be established, a tentative answer can be obtained by noting the operations count, defined in terms of multiplications and additions, involved in the two schemes.

In the case of the numerical approach, the number of independent nodes, n , and the number of branches, b , are the most important quantities for determining the computational effort along with the number of frequency points at which the output is desired. Assuming that no sparse matrix techniques are used, the numerical inversion of an $(n \times n)$ matrix requires $O(n^3/3)$ units of work, where $O(\) \equiv$ "order of", and 1 unit of work = one addition and one multiplication. For k frequency points, the work becomes $O(kn^3/3)$. This does not involve book-keeping and other pre- and post-processing steps such as pivoting and iterative refinement, which are usually necessary to insure reliability and robustness of the algorithm.

In the case of symbolic inversion using our approach, the important parameters in the computational effort are the dynamic degrees of freedom, d , and the number of ports, p , where voltages and currents are injected or measured. Using the QR algorithm [20,28] for computing the eigenvalues of

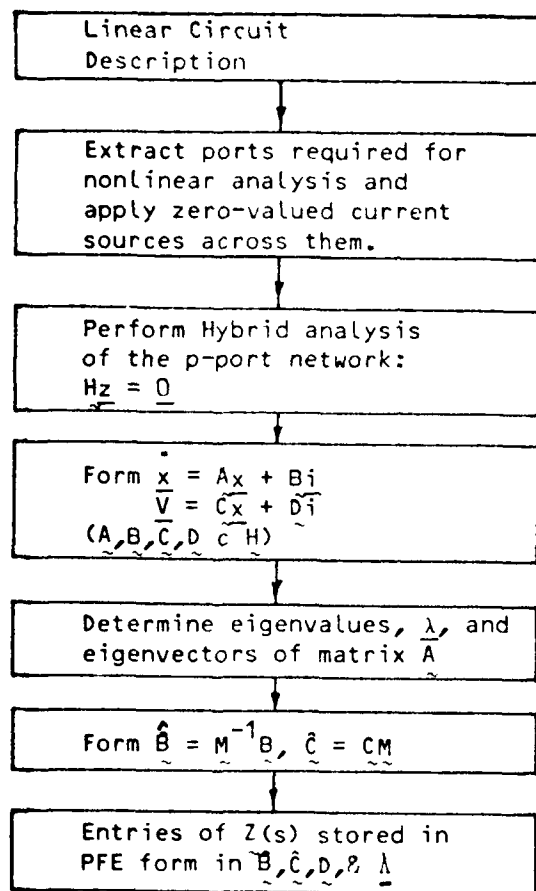


Fig. 3-2. Algorithm for inverting $\underline{Y}(s)$ symbolically.

the A matrix, the operation count is $O(8d^3)$. The total work required for obtaining the inverse at k frequency points is therefore $O(8d^3 + kdp^2)$. The number, p , depends on the number of nonlinearities in the circuit, and is usually small. Also, if the network complexity is less than the number of nodes, the symbolic approach would, in general, require less computational effort. As far as accuracy is concerned, both the QR algorithm and the Crout's algorithm with pivoting and iterative refinement yield accurate results.

The efficiency of the symbolic method rests heavily upon the availability on an efficient process for forming the state equations. The hybrid analysis method [19,20], which essentially reduces to the analysis of a resistive network, is well-suited for our purposes here.

3-4. Spectrum and Distortion Analysis Algorithm

The output spectrum and distortion indices for a nonlinear circuit with polynomial type nonlinearities can be computed on the basis of the material of Chapters 3 and 4. A flow-chart of the basic algorithm for such a computation is given in Fig. 3-3. We describe the steps involved in the following paragraphs:

Step 1: For the given nonlinear circuit, determine the dc operating point. Expand each nonlinear function into a Taylor series about the operating point to get a polynomial representation for the nonlinear element in terms of the incremental quantities. Thus, for example, a forward-biased diode having the "global" V-I representation

$$I = I_s [\exp(qV/nkT) - 1] \quad (3-15)$$

can be expanded into a Taylor series to yield the following incremental v-i representation:

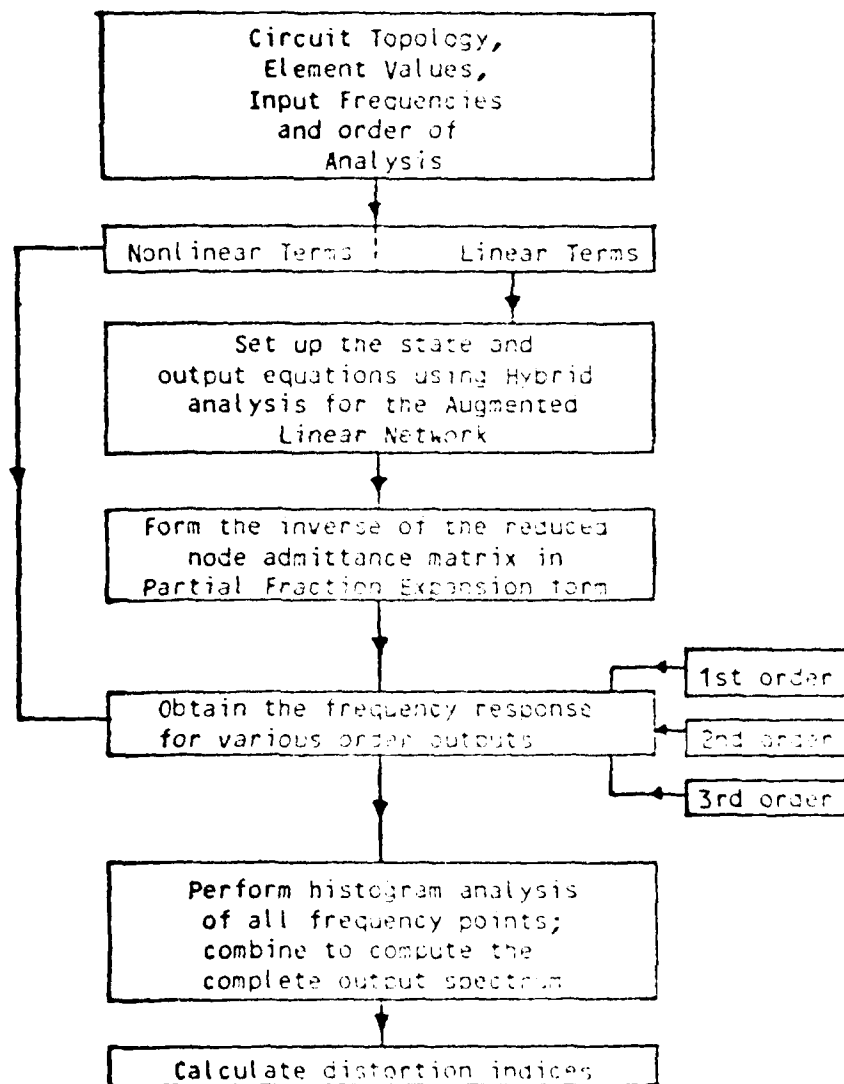


Figure 3-3. Algorithm for Spectrum and Distortion Analysis.

$$i = I_0 \frac{q}{mkT} v + \frac{I_0}{2!} \left(\frac{q}{nkT}\right)^2 v^2 + \frac{I_0}{3!} \left(\frac{q}{nkT}\right)^3 v^3 + \dots \quad (3-16)$$

where I_0 is the dc operating current.

Step 2: Lump the linear part of the nonlinear elements with the existing linear network to form the augmented linear network. Extract as ports the nonlinear element branches and the branches that control the nonlinear element characteristics (dependent nonlinear element case), along with the output and source branches, from the augmented linear network. Let $\underline{V} = [V_1 \ V_2 \ \dots \ V_p]$ and $\underline{I} = [I_1 \ I_2 \ \dots \ I_p]$ denote the vector of voltages and currents for these ports, respectively.

Step 3: Using a symbolic analysis algorithm (see Fig. 3.2), obtain the entries of the Z matrix as a function of s , where

$$\underline{V}(s) = \underline{Z}(s) \underline{I}(s) \quad (3-17)$$

For each of the input sources, and their associated frequency tones, compute the first-order output voltages at each of the extracted ports by using the appropriate entries of the Z matrix. This step amounts to letting $s = j\omega_i$ in $z_{ij}(s)$, the entries of $\underline{Z}(s)$.

Step 4: The second-order output spectrum is evaluated using the following relationship:

$$\underline{V}_2(s_1, s_2) = \underline{Z}(s_1 + s_2) \underline{I}_2(s_1, s_2) \quad (3-18)$$

The vector $\underline{I}_2(s_1, s_2)$ is the second-order current source vector, which is formed by using the coefficients associated with the quadratic term of the nonlinear element and the first-order output at the controlling port(s) of the nonlinearity. The latter information was obtained in step 3. The given input tones are taken two at a time in eqn. (3-18), along with the informa-

tion derived in Chapter 2, to evaluate the output voltages at each of the p-ports.

The third-order output spectrum is obtained in exactly the same manner. The first- and second-order outputs are used to form the third-order current source at each combination frequency, which is then pre-multiplied by evaluating $Z(s)$ at the combination frequency.

Step 5: Perform a histogram analysis of all frequency points and combine the responses at points which are repeated. The distortion indices are computed using:

$$HD_2 = \frac{|V_0(2\omega_i)|}{|V_0(\omega_i)|} \quad (3-19)$$

$$HD_3 = \frac{|V_0(3\omega_i)|}{|V_0(\omega_i)|} \quad (3-20)$$

where HD_2 and HD_3 denote the second and third order harmonic distortion indices.

3-5. Program PRANC.

The Program for Analysing Nonlinear Circuits, known as PRANC, is a digital computer program, written in FORTRAN IV, that computes up to the third-order complete output spectrum of a nonlinear circuit with polynomial nonlinearities driven by up to two multi-frequency inputs.* In the process it computes the Volterra transfer functions at each of the frequency combinations involved.

As mentioned previously, the solution of the nonlinear circuit problem reduces to the repeated solution of the linear circuit. To efficiently han-

*Thus, mixer-type circuits can be analyzed using PRANC.

dle this basic problem, PRANC uses a semi-symbolic approach [20] for analysing the augmented linear circuit. Specifically, the inverse of the reduced node admittance matrix is obtained in terms of the symbol s using the state equation formulation as described above.

The state equations for the linear circuit are formulated via the Hybrid analysis method [19,20]. If T denotes port branches in the tree [20] and C denotes port branches in the co-tree of a linear circuit, then the Hybrid analysis yields the following relationship:

$$\begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \\ H_{41} & H_{42} & H_{43} & H_{44} \end{bmatrix} \begin{bmatrix} i_T \\ v_C \\ v_T \\ i_C \end{bmatrix} = \underline{0} \quad (3-21)$$

$\underline{H} \qquad \underline{z}$

By suitably forcing the various ports in the linear circuit into the tree and co-tree branches, PRANC uses the (3-21) formulation for setting up the state equations. All capacitor branches are extracted as ports which necessarily become part of the tree and all inductors, nonlinear element branches (which are assumed to be voltage controlled), and input and output branches, are extracted as ports which are forced as part of the co-tree. The matrix \underline{H} is obtained in a form where $\underline{H}_{11} = \underline{I}$ (\underline{I} being the identity matrix), $\underline{H}_{12} = \underline{H}_{21} = 0$, $\underline{H}_{22} = \underline{I}$. This yields the capacitor currents and the inductor and nonlinear element branch voltages in terms of known variables. Thus, the A , B , C , and D matrices in the state and output equations (see eqns. 3-9 through 3-12) are obtained from the submatrices of \underline{H} . The formulation of eqn. (3-21) is quite fast, since it only involves the analysis of a resistive network.

It is noted that the matrix H may not exist in idealized circuits. However, for most practical circuit this matrix is almost certain to exist [20]. It should also be noted that the above formulation of state equations tacitly assumes that no degenerate cutsets (all inductor-current source cutset) or degenerate loops (all capacitor-voltage source loop) are present in the linearized circuit. These restrictions are not very severe, especially when the realistic lossy models of circuit components are taken into account.

The next step in the PRANC algorithm is to determine the eigenvalues and the eigenvectors of the A matrix. For this purpose, the double QR algorithm [28] for obtaining the eigenvalues is employed. The basic steps, such as matrix balancing, reduction to Hessenberg form, shift of origin, are included in this algorithm to make it efficient and reliable. The eigenvectors are also obtained in the process.

All information about the inverse of the reduced node admittance matrix is stored as three matrices and a vector. The matrices are \hat{B} , \hat{C} , and D (see eqns. 3-14), and the vector contains the eigenvalues. It is noted that the solution of eigenvectors for repeated eigenvalues can be a numerical unstable process [29]. Thus, the programs outputs a diagnostic message when such a case occurs.

The first-order voltage response at the prescribed ports is now computed from the entries of the open-circuit impedance matrix. These ports include: source port, output ports, nonlinear element ports, and ports which control the nonlinear element characteristics. The response is calculated for each user prescribed frequency, and stored as a two-dimensional array: port number vs. the frequency number.

The second-order voltage response is computed at each distinct combination of the input tones taken two at a time. The ports of interest are the same as that for the first-order response. The second-order current source vector, at a particular frequency combination, is formed by considering the nonlinear element type and the voltage(s) controlling it, which is determined from the first order response array. This vector is pre-multiplied by the open-circuit impedance matrix evaluated at the combination frequency to obtain the second-order transfer function vector at that frequency. The response voltage at this frequency is then determined from the transfer function value. The second-order transfer function values are again stored as a two-dimensional array: port number and the particular frequency combination.

The third-order response is determined similarly. The third-order current source vector is formed by properly picking out the values of the first- and second-order transfer functions. The indexing of the arrays is of critical importance to the efficient implementation of this scheme.

Since the hybrid analysis forms the basis for forming the open circuit impedance matrix, the following linear elements are allowed by the program*: resistors, capacitors, inductors, voltage or current sources, and all four types of controlled sources. The nonlinear elements are assumed to be voltage controlled, with the following polynomial descriptions:

$$i_p = a_1 f[v_p] + a_2 f[v_p^2] + a_3 f[v_p^3] \quad (3-22)$$

*A direct nodal analysis would only allow for voltage controlled current source.

$$i_p = a_{10}v_q + a_{01}v_r + a_{20}v_q^2 + a_{02}v_r^2 + \\ a_{11}v_qv_r + a_{30}v_q^3 + a_{03}v_r^3 + a_{12}v_qv_r^2 + a_{21}v_q^2v_r \quad (3-23)$$

where i_n and v_n are currents and voltages across branch n , f is a linear operator of the type $\frac{d}{dt}$, $\int_{-\infty}^t$, or constant, and a_{ij} are constants. It should be noted that eqn. (3-23) models a 3-port device.

In the present version, PRANC imposes the following restrictions on the circuit parameters: maximum number of elements (both linear and nonlinear) = 60; maximum number of nonlinear elements = 10; maximum number of dependent nonlinear elements (eqn. 3-23) = 5; maximum number of reactive elements = 20; maximum number of independent nodes = 30; number of input frequencies = 5. These restrictions can be relaxed if desired. The modular structure and algorithms of PRANC makes it possible to extend the order of analysis in a straightforward manner. The limit on the highest order will eventually be dictated by the storage restrictions of the computer.

The validity of the results obtained from using PRANC has been verified through hand-worked examples and with the results obtained from using NCAP [24]. In Chapter 4 we present examples showing the results obtained from the use of PRANC.

CHAPTER 4

USER'S GUIDE FOR PRANC

4-1. Introduction

Based upon the theory of Chapter 2 and the algorithms of Chapter 3, PRANC (Program for Analyzing Nonlinear Circuits), a digital computer program, has been developed for the sinusoidal steady state analysis of circuits with multiple nonlinear elements and multiple multi-frequency input sources. The complete listing of the program is contained in Chapter 5.

The usefulness of PRANC is not restricted only to users who are well-versed in the Volterra series method; users with a basic knowledge of the significance of sinusoidal steady state analysis, eigenvalues (poles) of a linear system, and other related circuit analysis concepts can easily use the program, and understand the information provided by it. By suitably translating the circuit analysis problem into a prescribed sequence of well-defined statements - to be presented in this chapter - any user can use PRANC as an analysis tool.

To methodically and effectively use PRANC, the user is recommended to follow a three-step process:

1. Preliminary Data Preparation
2. Translation of Data for Analysis
3. Interpretation of Analysis' results.

The contents of this Chapter are organized on the basis of these steps.

In section 4-2, the considerations entailed in the preliminary data preparation are presented. The allowed elements, the user available options, and the program restrictions in terms of the circuit size and features are discussed.

Section 4-3 presents the sequence of input cards (input data to the program) for PRANC. The formats for each card in the sequence is described. The interpretation of the program output is the subject of section 4-4. Finally, several examples are presented in section 4-5 to illustrate the use of PRANC.

4-2. Preliminary Data Preparation

4-2.1 Allowable Element Types: The first step in any circuit analysis problem is the drawing of its complete circuit model. This diagram should include all elements which can be identified by PRANC.

The present version of PRANC is capable of identifying the following element types, which are depicted in Fig. 4-1:

- Independent voltage source
- Linear Components: Resistor, Inductor, and Capacitor
- Linear Dependent Sources: Voltage-controlled Voltage source, Current-Controlled Current-Source, Voltage-controlled current-source, and Current-controlled voltage-source.
- Nonlinear Components: Resistor, Inductor, and Capacitor
- Nonlinear Dependent Source: Voltage-controlled current-source.
- Bipolar Junction Transistor

The polarity convention assumed by PRANC is shown in Fig. 4-1. The current voltage relationships for linear elements are well-known. The nonlinear elements are assumed to be represented in the form of polynomials of branch voltage(s). Thus, PRANC handles nonlinear elements expressed as:

Element	Symbol
<u>Linear Components:</u> Resistor	
Capacitor	
Inductor	
<u>Linear Dependent Sources:</u> Voltage-Controlled Voltage Source	
Voltage-Controlled Current Source	
Current-Controlled Voltage Source	
Current-Controlled Voltage Source	
<u>Nonlinear Components:</u> Resistor	

Figure 4-1 PRANC Element Definitions
63

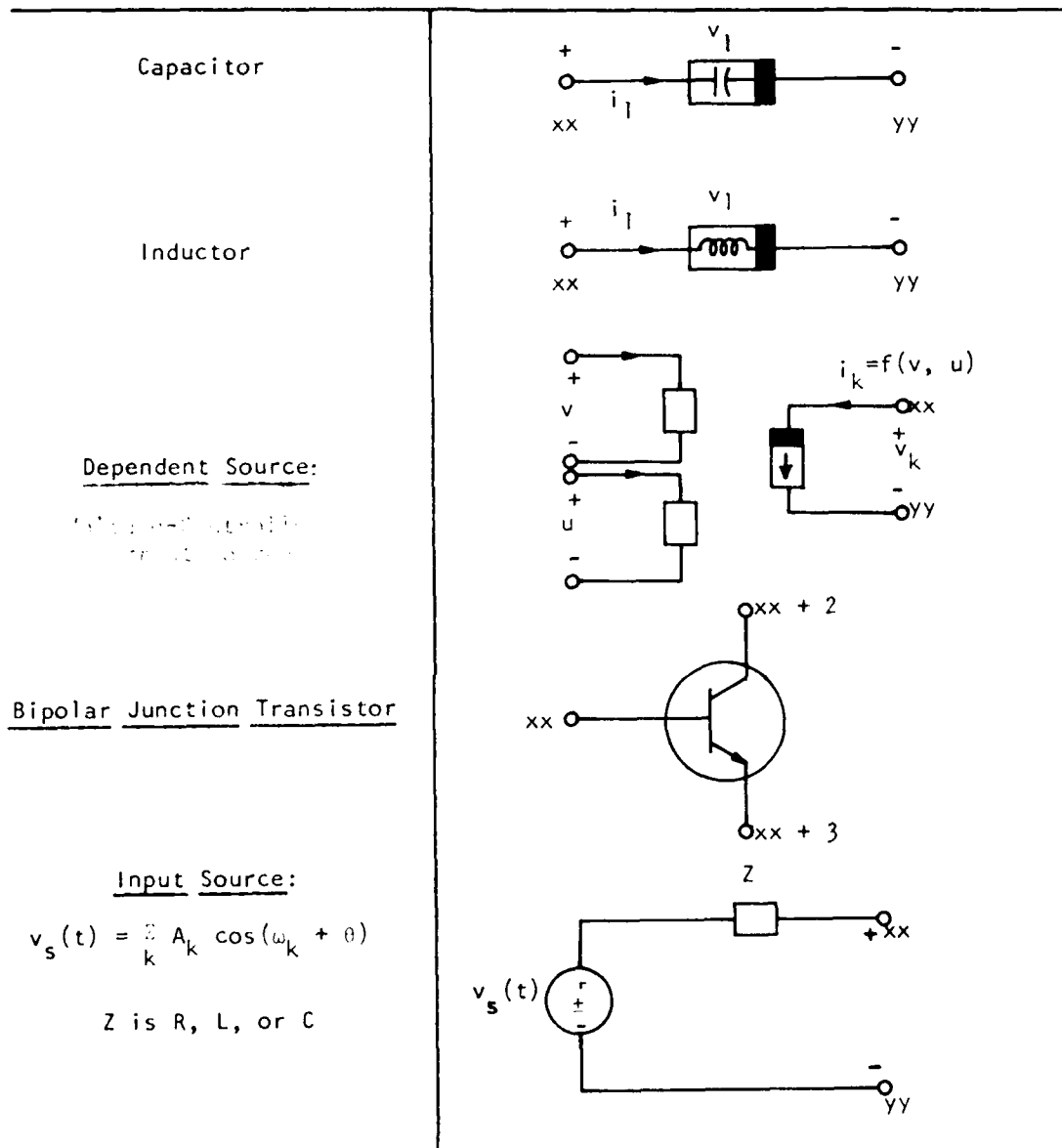


Figure 4-1 (Contd.) PRANC Element Definitions

$$i_{NL} = a_1 f[v] + a_2 f[v^2] + a_3 f[v^3] \quad (4-1)$$

or

$$i_{NL} = a_{10}v + a_{01}u + a_{20}v^2 + a_{02}u^2 + a_{11}vu \\ + a_{30}v^3 + a_{03}u^3 + a_{21}v^2u + a_{12}vu^2 \quad (4-2)$$

where f is an operator of the form $\int, \frac{d}{dt}$, or a constant, u and v are branch voltages, and i_{NL} is the current across the nonlinear element. It should be noted that eqn. (4-1) is adequate to model a nonlinear capacitor, a nonlinear inductor, or a nonlinear resistor, and that eqn. (4-2) is suitable to model a 3-port or a 2-port voltage controlled nonlinear dependent source.

The representation of a nonlinear device in terms of a polynomial is covered in several papers and reports [7,10]. An example of the development of a polynomial representation for a semiconductor diode is given in Appendix A.

It should be noted that if a current-controlled nonlinear element is present in the circuit, the reversion of the series may be used. That is, given

$$v_{NL} = a_1 i_{NL} + a_2 i_{NL}^2 + a_3 i_{NL}^3 \quad (a_1 \neq 0) \quad (4-3)$$

We can express

$$i_{NL} = A_1 v_{NL} + A_2 v_{NL}^2 + A_3 v_{NL}^3 \quad (4-4)$$

where

$$A_1 = \frac{1}{a_1} \quad (4-5)$$

$$A_2 = -\frac{a_2}{a_1} \quad (4-6)$$

$$A_3 = \frac{1}{a_1} (2a_2^2 - a_1 a_3) \quad (4-7)$$

where i_{NL} and v_{NL} are the current and voltage across the nonlinear element, respectively.

The element node numbers are shown by symbols xx and yy in Fig. 4-1. For devices representable in terms of a pair of nodes, or a collection thereof, the node numbers are assigned by the user. The node numbering for a bipolar junction transistor is done internally within the program once the node number for the base terminal of the transistor has been specified by the user. The model for the transistor used in PRANC is based on Narayanan's work [7], and is shown in Fig. 4-2 along with the program-assigned node numbers.

4-2.2. User Available Options:

PRANC performs the complete sinusoidal steady state analysis of a non-linear circuit. In accomplishing this task, the program obtains the state equations and the eigenvalues for the linearized circuit, forms the entries of the open circuit impedance matrix* in partial fraction expansion form, and then computes the first-, second-, and third-order transfer function values at each combination of the positive and negative input frequency values. The output voltage at each frequency component is then computed

*This is an equivalent form of the inverse of the reduced node admittance.

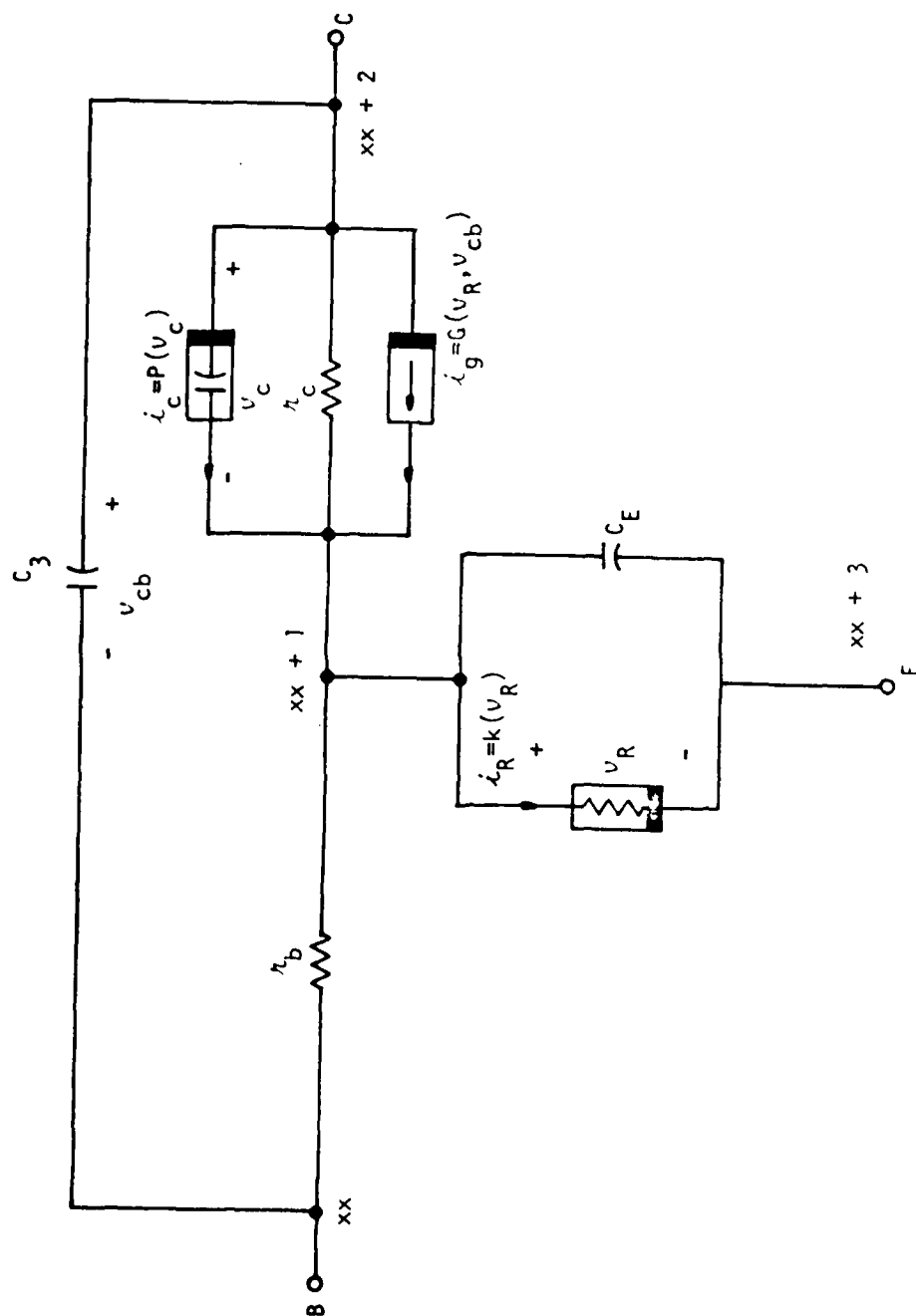


Figure 4-2. Transistor Equivalent Model.

from the transfer function. The sinusoidal steady state response is obtained after combining the various order responses at repeated frequency values.

In order to provide flexibility to the user to control the program output, several options have been incorporated in PRANC. These are described next.

1. Frequency Sweep. Many applications in distortion and spectrum analysis of nonlinear circuits calls for the study of the effect of frequency on the distortion products. A frequency sweep capability, which allows the user to request multiple analyses of a given circuit over a range of generator frequency values in a single execution, is provided by PRANC. This option can be called for by specifying the acronym FS on the option card.

PRANC allows the user to sweep up to five tones*. This allows the user to study the effect of frequency on the second- and third-order intermodulation products independently. In a third order analysis, a combination of three input frequencies is taken at a time to compute the amplitude of an intermodulation product; sweeping up to three frequencies is therefore sufficient to study the effect of frequency on an intermodulation product. Thus, given a fixed intermodulation frequency $\omega_{IM} = \omega_1 + \omega_2 - \omega_3$, where ω_1 , ω_2 , and ω_3 are the input frequencies, the effect of a change in the input frequencies on ω_{IM} can be investigated by simultaneously incrementing ω_1 , ω_2 , and ω_3 by a fixed amount across the band of interest. The study of the effect of frequency on a second-order distortion product is done similarly by varying two tones at a time. Both linear and logarithmic frequency sweeps are available on PRANC.

*Since PRANC generates the negative of the input frequencies internally, this is equivalent to sweeping ten frequencies.

2. Multiple Input Sources. Ordinarily PRANC assumes the nonlinear circuit to have a single multi-frequency input source. However, when a two source circuit, such as a mixer circuit, is to be analyzed, the acronym MX (mixer) on the option card can be used. PRANC will, in such a case, look for the description of the second input source. The first generator can have up to four input tones, and the second generator (the "local oscillator") only one input frequency.

3. Print and Plot Complete Spectrum. After computing the transfer functions and output voltages, PRANC performs a histogram type analysis of all frequency points to compute the complete output spectrum across a requested circuit element for printing and plotting purposes. Often times the user may be only interested in the transfer function and output voltage values, and may have no use for the complete output spectrum. In order to provide the flexibility for suppressing the printing and plotting of the complete output spectrum, an option to be specified by the user is available. By using the acronym PC on the option card the user can request for a print-out and plot of the complete output spectrum; an absence of PC on the option card signals the program to suppress the histogram analysis feature.

4. Output Port Print-out Selection. PRANC performs the analysis of the nonlinear circuit on a port basis. Two types of ports are extracted in the analysis: 1) Input and output ports specified by the user, and 2) controlling ports for the nonlinear elements. Depending on the number of nonlinear elements and the number of controlling voltage variables, the number of the extracted ports can become quite large and thereby result in an inordinately large amount of printed output if the transfer function and output voltage at each frequency component and at each of the ports is printed. To reduce

the amount of printed output, an option for the printing of selected output ports can be requested. By using the acronym AP (All extracted ports) on the option card, the transfer functions and output voltages at each of the extracted ports is printed; an absence of AP on the option card signals the program to print the transfer functions and output voltages at only the user-prescribed output ports.

5. State Space Description Print. The open circuit impedance matrix for the linearized circuit is obtained via the state space description (see eqn. 3-5). The user can request a print-out of this description by specifying the acronym SE on the option card. When SE is omitted from the option card, the printing of the \underline{A} , \underline{B} , \underline{C} and \underline{D} matrices is suppressed.

6. Eigenvalue, Modal Matrix Print. The eigenvalues or the poles of the linearized circuit, and their associated eigenvectors, are computed by PRANC. The user may access this information by specifying NM (natural modes) on the option card. The eigenvalues and the modal matrix are not printed when the letters NM are omitted from the option card.

7. Open Circuit Impedance Matrix Print. The open circuit impedance matrix for the linearized circuit is obtained in partial fraction expansion form by PRANC. Each entry of this matrix is obtained in terms of a set of pole-residue pairs and can be written as:

$$z_{ij}(s) = \sum_k \frac{r_k}{s - \lambda_k} + \text{constant} \quad (4-8)$$

where r_k is the residue associated with the pole λ_k . Knowing all the entries of the open-circuit impedance matrix in the form (4-8), it is possible to obtain the higher order transfer functions in terms of s_i [23]. By using

the acronym PR (pole-residue) on the option card, all information required to obtain each entry of the open circuit impedance matrix in the form (4-8) can be accessed from PRANC.

8. Debug Print. The hybrid analysis formulation is used by PRANC to set up the state space description of the linearized circuit. All important intermediate results leading to the determining of the hybrid matrix can be obtained by the user by requesting a debug run. This option is invoked by specifying the acronym DB on the option card.

4-2.3. Program Restrictions

The present version of PRANC imposes the following restrictions on the circuit size:

Maximum number of elements (both linear and nonlinear) =	60
Maximum number of nonlinear elements ⁺ =	10
Maximum number of dependent nonlinear elements =	5
Maximum number of reactive elements =	20
Maximum number of independent nodes =	30
Maximum number of input frequencies* =	5
Maximum number of extracted ports** =	25
Maximum number of inputs =	2

In addition to the above size restrictions, there are other restrictions imposed by the algorithms used: the presence of degenerate (all capacitor-voltage source) loops or degenerate (all inductor-current source)

⁺A bipolar transistor accounts for three nonlinear elements.

*These are the sine wave input frequencies. The negative frequencies are generated within the program.

**Number of extracted ports $\leq NO + NINL + 3NDNL + 1$;

NO \equiv number of requested outputs

NINL = number of independent nonlinear elements

NDNL = number of dependent nonlinear elements.

cutsets [20] will lead to erroneous results. It should be noted that this restriction is not severe when the realistic lossy models for capacitors or inductors are used. A series resistance with a capacitor or a shunt resistance with an inductor to account for the element non-idealities will insure the absence of any of the aforementioned degenerate cases.

Another restriction encountered in PRANC is related to the determination of the eigenvectors. It is well known [29] that the computation of the eigenvectors for repeated eigenvalues can be an ill-conditioned problem. Thus, whenever a linearized circuit has repeated eigenvalues, PRANC outputs a diagnostic message*. Again it is remarked that this restriction is not very severe. One can easily concoct simple network examples with repeated eigenvalues; but in real-life circuits, the probability of encountering repeated eigenvalues is very low - particularly when one considers the method of storing numbers in the finite length word of any digital computer.

To summarize this sub-section, the following three-step procedure is recommended to the user as part of the preliminary data preparation:

Step 1: Examine the circuit under consideration to insure that all elements are recognizable by PRANC. Furthermore, insure that there are no degenerate loops or degenerate cutsets [20]. If such conditions exist, the following remedy is recommended: place a negligibly small resistor in a degenerate loop; place a negligibly small conductance in parallel with one of the elements of the degenerate cutset.

Step 2: Assign consecutive numbers to all elements in the circuit (including a bipolar transistor) from 1 to NB and all nodes in the circuit from 0 to NN, where NB is the number of elements (both linear and nonlinear) and NN is

*It should be noted that this is due only to the numerical problems and that the theory of chapters 2 and 3 is still valid.

the number of independent nodes. Node number 0 is assumed to be the ground node. Insure that the circuit size does not exceed the limits imposed by the present version of PRANC.

Step 3: Note the number of linear and nonlinear elements, and the number and unit of the input frequencies. Based on the list of available options, select the ones desirable for the circuit analysis problem at hand.

4-3. Input Description for PRANC

In this sub-section, the details of the prescribed sequence of cards needed for using PRANC are presented. After the preliminary data preparation step is done, the procedure for translating the circuit description into input data is straightforward.

Assuming that PRANC is stored in the computer, the sequence of cards needed for the analysis of a nonlinear circuit with a single source is shown in Fig. 4-3; the case of the two-input source circuit is shown in Fig. 4-4. There are basically six types of cards present in the input data for PRANC. These are: 1) Title card; 2) Option Card; 3) Analysis Parameter card; 4) Linear Component description cards; 5) Nonlinear component description cards; and 6) Generator description cards. The details of the contents of each of these card types are described next.

1. Title Card: This card is read in with an 80A1 form and is reproduced as the first line of the output.

2. Options Card: This card tells the program which options, described in section 4-2.2, are desired by the user. Each option has a two-letter acronym associated with it. These acronyms are summarized in Table 4-1. Starting in column 1 the user must punch a contiguous string of the acronyms re-

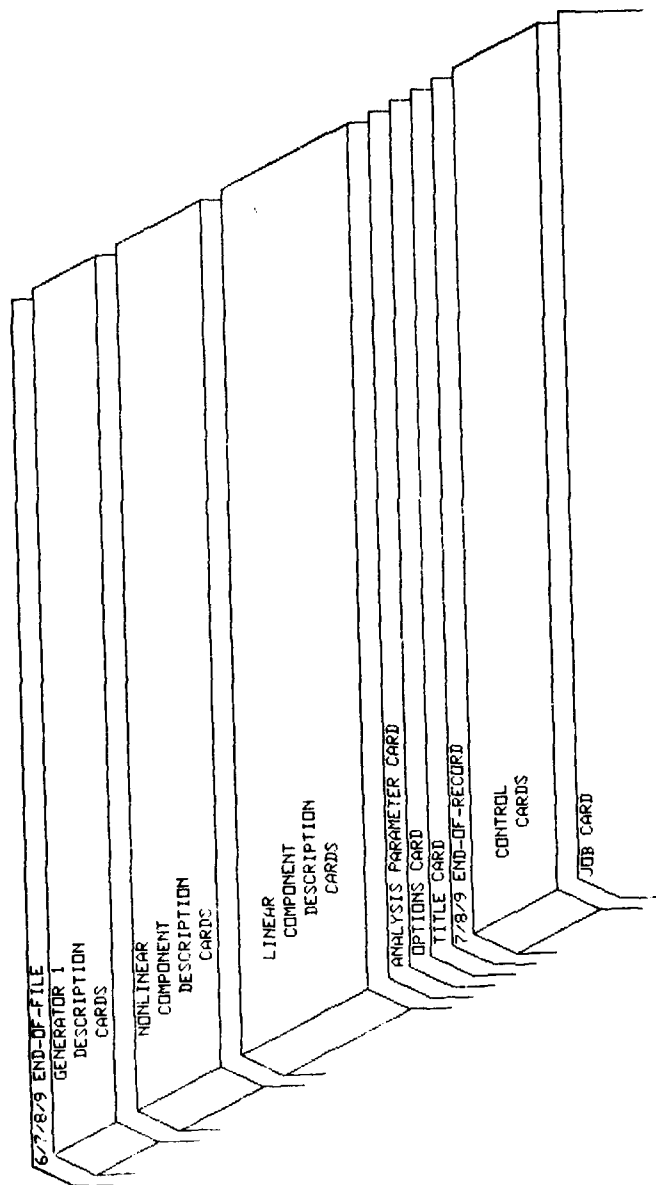


Figure 4-3. Sequence of Cards for Single Input Circuits

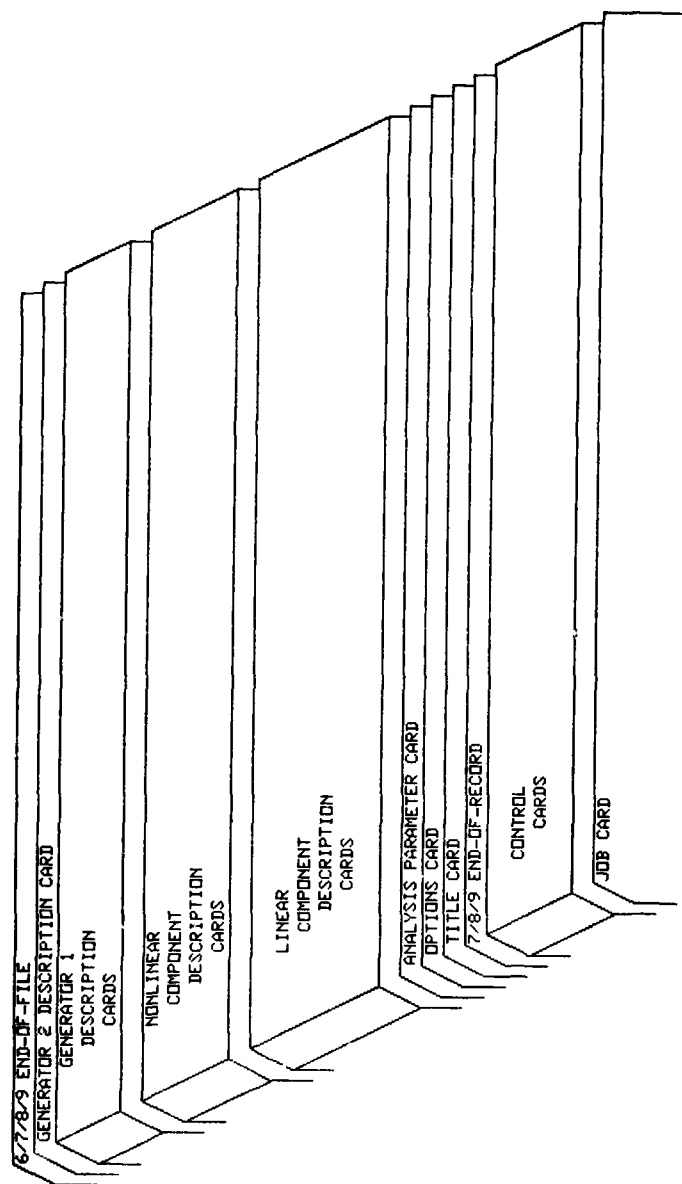


Figure 4-4. Sequence of Cards for Two-Input Circuits

quired to request the specific options. The card must therefore be in the following format:

<u>Column</u>	<u>Format</u>	<u>Description</u>
1-2	A2	First desired option acronym
3-4	A2	Second desired option acronym
5-6	A2	Third desired option acronym
.	.	
.	.	
.	.	

See section 4-5 for examples.

3. Analysis Parameter Card: The analysis parameter card contains information regarding the number of linear elements, the number of nonlinear elements (the transistor should be counted as 1 nonlinear element by the user), the number of sinusoidal frequencies (≤ 5) in the input signal*, the unit of the input frequencies, and the type of frequency sweep (if desired). This card must be in the following format:

<u>Column</u>	<u>Format</u>	<u>Description</u>
1-2	I2	No. of linear elements
3-4	I2	No. of nonlinear elements
5	I1	No. of input frequencies
6-8	A3	Unit for the input frequencies use RAD for rad/sec. Hz for Hertz
9-11	A3	Type of frequency sweep, if desired; LIN for linear and LOG for logarithmic

*This does not include the local oscillator frequency in the case of a mixer circuit.

Table 4-1. Summary of Available User Options on PRANC

Option Acronym		Option Description
1.	AP	Print all extracted ports output information
2.	DB	Debug for Hybrid analysis: print all intermediate results
3.	FS	Frequency sweep capability
4.	MX	Two-input circuit for analysis
5.	NM	Print eigenvalue and modal matrix information
6.	PC	Print and plot complete output spectrum
7.	PR	Print pole-residue information for the open-circuit impedance matrix
8.	SE	Print state-space description of the linearized circuit

It should be noted that the I-format and the A-format are always right justified.

4. Linear Component Description Cards: Each branch containing a linear element save the independent source(s) and its impedance(s) must be described in terms of its topological connections, its element value and type, and its controlling branch number, if any. Each linear element description card must use the following format:

<u>Column</u>	<u>Format</u>	<u>Description</u>
1-3	I3	Branch number
4-6	I3	Positive ("From") node number
7-9	I3	Negative ("To") node number (Sign convention for PRANC is shown in Fig. 4-1)
10-11	A2	Element type. The following element types and their acronyms are recognized by PRANC: R Resistance G Conductance L Inductance C Capacitance CV Current-controlled Voltage source VV Voltage-controlled Voltage source VC Voltage-controlled Current source CC Current-controlled Current source (Note: R, L, G, or C must be present in column 11: right-justified)
12-21	E10.3*	Element value of R, G, L, C, or dependent source.
22-24	I3	Branch number for the controlling branch of CC, CV, VV, or VC. For other element

*(Note: For an E-format input: the exponent appears as a signed two digit integer in the three right most columns of the format field, and is preceded by a letter E, which is preceded by the floating point value. Thus, for example, a 6.6 μ F capacitor value should appear as:

12 13 14 15 16 17 18 19 20 21)
1 6 . 6 0 0 E - 0 6).

types this should be left blank.

- | | | |
|----|----|--|
| 25 | I1 | This column is used to provide multiple output capability. A 1 in this column indicates that the current branch number is an output branch; a blank indicates otherwise. |
| 26 | A1 | An asterisk (*) in this column indicates that the complete output spectrum across this branch should be printed and plotted; a blank indicates otherwise. <u>Note</u> : only one such branch is allowed in the current version of PRANC. |

5. Nonlinear Component Description Cards: Two cards are used to describe each nonlinear capacitor, inductor, resistor, or a dependent source. The first card describes the nonlinear component type and its connection in the circuit; and the following card, the second card, defines the coefficient values in the polynomial expansion of the nonlinear element (as per eqns. 4-1 and 4-2). The nonlinear components are assumed to have a voltage-controlled current.

The format for the two cards required to define a two terminal nonlinear component is as follows:

<u>First Card:</u>		<u>Description</u>
<u>Column</u>	<u>Format</u>	
1-3	I3	Component number
4-6	I3	Positive ("from") node number of the component
7-9	I3	Negative ("to") node number of the component
10-11	A2	Element Type. The following acronyms are allowed for the various element types: NC Nonlinear Capacitor NL Nonlinear Inductor NR Nonlinear Resistor ND Dependent Nonlinearity (eqn. 4-2)
12-14	I3	First controlling voltage branch

number for the dependent nonlinearity.
These columns are left blank in the case
of NC, NL, or NR.

15-17 13 Second controlling voltage branch number
for the dependent nonlinearity. These
columns are left blank in the case of
NC, NL, NR, or single-voltage-controlled
dependent nonlinearity.

Second Card: This card is used to define the coefficients of the polynomial
describing the nonlinear element described on the first card. The format
for this card is:

<u>Column</u>	<u>Format</u>	<u>Description</u>
1-10	E10.3	Coefficient a_1 in eqn. (4-1) or coefficient a_{10} in eqn. (4-2)
11-20	E10.3	Coefficient a_2 in eqn. (4-1) or coefficient a_{01} in eqn. (4-2)
21-30	E10.3	Coefficient a_3 in eqn. (4-1) or coefficient a_{20} in eqn. (4-2)
31-40	E10.3	Coefficient a_{02} in eqn. (4-2)
41-50	E10.3	Coefficient a_{11} in eqn. (4-2)
51-60	E10.3	Coefficient a_{30} in eqn. (4-2)
61-70	E10.3	Coefficient a_{03} in eqn. (4-2)
71-80	E10.3	Coefficient a_{21} in eqn. (4-2)
1-10 (new card)	E10.3	Coefficient a_{12} in eqn. (4-2)

Three cards are needed to describe each bipolar transistor in the circuit.
The first card indicates that a transistor is present and also specifies
the node number for the external base terminal. The second and third
cards input the transistor parameters for the purpose of PRANC modelling.
These parameters include the following (see Fig. 4-2):

<u>Parameter No.</u>	<u>Parameter Name</u>	<u>Description</u>
1	n	Avalanche Exponent

2	V_{cB}	Collector-base bias Voltage
3	V_{cB0}	Avalanche Voltage
4	μ	Collector Capacitance Exponent
5	I_c	Collector bias current
6	I_{cmax}	Collector current at maximum d.c. current gain
7	a	h_{FE} nonlinearity coefficient
8	h_{FEmax}	maximum d.c. current gain
9	k	collector capacitance scale factor
10	Ref	Diode non-ideality factor
11	C_{je}	Base-emitter junction space charge capacitance
12	C_2'	Derivative of base-emitter diffusion capacitance
13	r_b	Base resistance
14	r_c	Collector resistance
15	C_1	Base-emitter capacitance
16	C_3	Base-collector and overlap capacitance

Once the external base terminal node, xx, has been specified by the user, the following node numbers are internally assigned by the program to the other terminals in the transistor model:

xx + 1 : Internal Junction
xx + 2 : External collector
xx + 3 : External emitter

The user must therefore take caution in not assigning these node numbers elsewhere in the circuit.

For the bipolar junction transistor description, the following sequence of cards are used:

First Card: The format of the first card for the description of a BJT is identical to that for the two terminal nonlinear components. Accordingly, the following format is used:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-3	I3	Component number*
4-6	I3	External base node number
7-9	I3	(blank)
10-11	A2	The acronym TR in these columns signals the presence of a bipolar junction transistor.

A TR in columns 10-11 on the first card of the nonlinear description cards causes PRANC to read two additional cards describing the transistor parameters. The format and the order in which the parameters are read is as follows:

<u>Second Card:</u>			
<u>Columns</u>	<u>Format</u>		<u>Description</u>
1-10	E10.3	n	: Avalanche Exponent Value
11-20	E10.3	V_{CB}	: Collector-base bias voltage value
21-30	E10.3	V_{CB0}	: Avalanche voltage value
31-40	E10.3	μ	: Collector capacitance exponent value
41-50	E10.3	I_C	: Collector bias current value
51-60	E10.3	I_{Cmax}	: Collector current at maximum d.c. gain value
61-70	E10.3	a	: h_{FE} nonlinearity coefficient value
71-80	E10.3	h_{FEmax}	: maximum current gain value

*Each transistor should be counted as one nonlinear component in the circuit.

Third Card:

<u>Columns</u>	<u>Format</u>	<u>Description</u>	
1-10	E10.3	k	: collector capacitance scale factor value
11-20	E10.3	Ref	: diode non-ideality factor value
21-30	E10.3	C_{je}	: Base-emitter junction space-charge capacitance value
31-40	E10.3	C_2'	: Derivative of base-emitter diffusion capacitance value
41-50	E10.3	r_b	: base resistance value
51-60	E10.3	r_c	: collector resistance value
61-70	E10.3	C_1	: base-emitter capacitance value
71-80	E10.3	C_3	: base-collector and overlap capacitance value

In summary, the nonlinear component description cards are a sequence of cards where:

- 1) Two cards are used to describe each nonlinear resistor, nonlinear capacitor, or nonlinear inductor;
- 2) Three cards are used to describe each nonlinear dependent source;
- 3) Three cards are used to describe each bipolar junction transistor in the circuit.

6. Generator Description Cards: PRANC assumes the independent source to be a voltage source in series with an impedance, as shown previously in Fig. 4.1. The impedance can be a linear resistor, a linear capacitor, or a linear inductor. Two types of cards are required to describe the generator: the first card specifies the generator connection in the circuit and the succeeding cards describe the frequencies and their associated amplitudes along with the parameters for frequency sweep capability, if desired by the user.

Only two nodes are needed to specify the connection of the generator to the circuit.

The input voltage source is assumed to have the following form:

$$v_s(t) = \sum_{i=1}^n A_i \cos(\omega_i t + \theta_i); n \leq 5 \quad (4-9)$$

The user is therefore required to input the values for A_i , ω_i , and θ_i to describe the input source.

When the frequency sweep capability is requested by the user on the option card, the following three quantities must also be specified along with A_i , ω_i , and θ_i : 1) the number of steps or frequency increments; 2) the highest or terminal value of the frequency sweep; and 3) type of the desired sweep, which indicates whether the increment is to be linear (additive) or logarithmic (multiplicative).

It should be noted that the number of steps defines the number of times the circuit is to be analyzed. For linear sweeps the value of the increment is calculated by the program according to the expression:

$$INC_i = \frac{HFR_i - FR_i}{NSTP_i - 1} \quad (4-8)$$

where $INC_i \equiv$ frequency increment value for the i -th frequency,

$HFR_i \equiv$ highest value for the i -th frequency,

$FR_i \equiv$ starting value for the i -th frequency,

$NSTP_i \equiv$ number of increments for the i -th frequency,

Similarly, for a logarithmic sweep, the increments are calculated as follows:

$$INC_i = \left[\frac{HFR_i}{FR_i} \right] \frac{1}{NSTP_i - 1} \quad (4-11)$$

(4-12)

In determining the value for the number of increments, the user should be aware that the highest and the starting frequency values each count as an increment. It should also be noted that multiple frequency sweep specifications always result in simultaneous increments of the frequency values involved. The largest defined "number of increments" value determines the number of analyses to be performed in such cases. As the analysis progresses, each frequency value will be incremented until its highest value has been reached, after which it will remain constant until all defined frequency sweeps have been satisfied.

The first card in the generator description card has the following input format:

<u>Column</u>	<u>Format</u>	<u>Description</u>
1-3	I3	Positive ("from") node number for the generator
4-6	I3	Negative ("To") node of the generator
7-8	A2	Source Impedance Type: <u>R</u> , <u>L</u> , or <u>C</u>
9-18	E10.3	Source impedance element value

The cards following the above card provide information about each frequency value, along with its associated amplitude and phase, and its frequency sweep parameters. The format used to describe the i-th input frequency is as follows:

<u>Column</u>	<u>Format</u>	<u>Description</u>
1-10	E10.3	Amplitude value for the i-th frequency
11-20	E10.3	i-th input frequency value (must be greater than 0)
21-30	E10.3	Phase value in degrees for the i-th frequency
31-40	E10.3	Highest value for the i-th input frequency. Should be left blank when frequency sweep capability is not desired.
41-42	I2	Number of increments desired for the i-th frequency.

When two input sources are present in the circuit being analyzed, and the acronym MX has been included on the option card, the card immediately following the above "frequency description" cards is used to define the second source ("local oscillator") parameters. The second source is again assumed to be a voltage source with a series impedance of the resistive, inductive, or capacitive type. Only one frequency value is however allowed for the second source. The description of the second source must have the following format:

<u>Column</u>	<u>Format</u>	<u>Description</u>
1-3	I3	Positive ("from") node number for the source
4-6	I3	Negative ("to") node number for the source
7-8	A2	Source impedance type: R, L, or C
9-18	E10.3	Source impedance element value
19-28	E10.3	Amplitude value of source
29-38	E10.3	Source ("local oscillator") frequency value
39-48	E10.3	Phase value in degrees for the source

In section 4-5 we shall present concrete examples to illustrate the typical sequence of cards used to translate nonlinear circuit problems for analysis using PRANC.

4-4. Interpretation of PRANC Output

A typical PRANC output comprises a large volume of printed information. In general, even when all user available options are suppressed, the output consists of: 1) images of all input cards; 2) all circuit devices* with their associated parameters and polynomial representation of their nonlinearities; 3) the description of the augmented linear network; 4) the description of all extracted ports; and 5) the transfer functions and output voltages across the desired output ports. The transfer functions and the output voltages are printed for all non-negative ("positive" frequency spectrum) combinations of every positive and negative input sinusoidal frequencies**, in both cartesian and log polar form. Thus, if a two-tone generator is specified by the user, with $2f_2 > f_1 > f_2$, PRANC will print the transfer function and output voltage values at the following frequency combinations:

First order : f_1, f_2

Second order : $2f_1, f_1+f_2, f_1-f_2, 0, 2f_2$

Third order : $3f_1, 2f_1+f_2, f_1, 2f_1-f_2, f_1+2f_2, 3f_2, 2f_2-f_1, f_2$

When the available user options are used, additional information about the circuit is provided by PRANC. The details of each of the available option was presented in section 4-2. We briefly repeat their functions here.

*These include bipolar junction transistor parameters in the present version.

**The user specifies only the positive sinusoidal frequencies; PRANC generates their negative values within the program.

When the acronym SE is punched on the option card, PRANC will print the complete state space formulation for the augmented linear network. It is well-known [20] that, like the nodal or loop analysis, a linear network is completely characterized by its state space description. By isolating the dynamic (energy storage) elements in the linear network, the state equation description emphasizes the dynamic character of the linear part of the non-linear circuit under study. PRANC isolates the capacitor voltages and the inductor currents as the state variables for the linearized network. It prints the A, B, C, and D matrices of the following vector equations:

$$\begin{aligned}\dot{\underline{x}} &= \underline{A}\underline{x} + \underline{B}\underline{i} \\ \underline{y} &= \underline{C}\underline{x} + \underline{D}\underline{i}\end{aligned}\tag{4-13}$$

where the vector $\underline{x} = [v_{c1} \ v_{c2} \dots v_{cn} \ i_{L1} \ i_{L2} \dots i_{Lk}]^T$,

the vector $\underline{i} = [i_1 \ i_2 \dots i_p]^T$,

and the vector $\underline{y} = [v_1 \ v_2 \dots v_p]^T$.

Here v_{ci} is the i-th capacitor voltage, i_{Li} is the i-th inductor current, and v_k and i_k are the voltages and currents for the k-th extracted port, respectively. The order in which the states are arranged is identical to the order in which the capacitors and inductors appear in the augmented linear description, which is always printed by PRANC in a typical successful execution of the program.

When the acronym NM appears on the option card, the eigenvalues (poles) and the modal matrix for the augmented linear network is printed. The significance of this information is well-known [20]: the poles have a direct bearing on the linear system response and stability; the modal matrix can be used to study the zero-input response along with the observability and controllability properties of the linearized system.

AD-A088 422

PURDUE UNIV LAFAYETTE IND

F/8 9/3

PRANC: PROGRAM FOR ANALYZING NONLINEAR CIRCUITS.(U)

MAY 80 H K THAPAR, B J LEON

F30602-78-C-0102

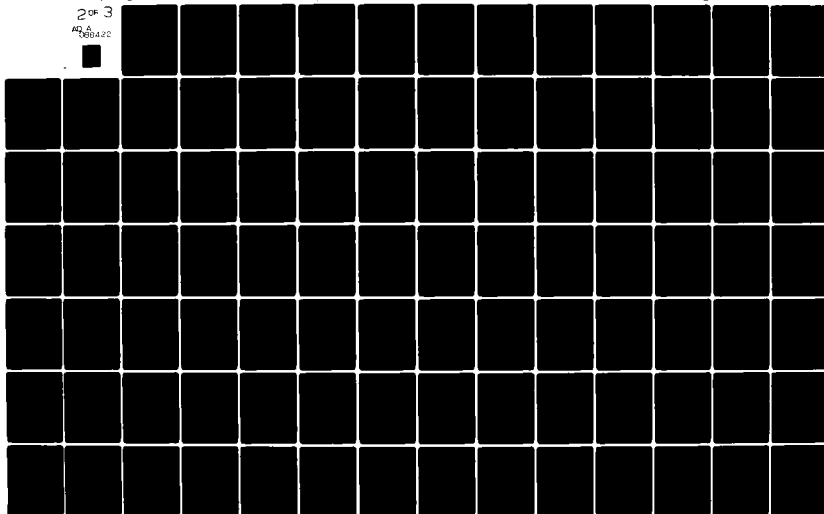
UNCLASSIFIED

RADC-TR-80-139

NL

2 OF 3

AD-A088 422



The presence of the acronym PR on the option card causes PRANC to print the pole-residue information of each entry of the open-circuit impedance matrix for the p-port augmented linear circuit. This information can be used to construct the higher order transfer functions in terms of the transform variables s_i . Multi-dimensional transform theory [5] can then be applied to these transfer functions to get more insight into the operation of the nonlinear circuit.

The presence of the acronym AP on the option card causes PRANC to print the transfer function and the output voltage values for all the ports extracted for analyzing the nonlinear circuit problem. These ports include: 1) input source port(s); 2) user requested output ports; 3) the ports at which the nonlinear elements are present; and 4) the ports which control the nonlinear element characteristics.

When the acronym PC is present on the options card, the complete steady-state response at the "most desirable", user-specified output port is obtained and printed. The logarithm of the output voltage is also plotted as a bar-graph, which has the same display characteristic as a spectrum analyzer. As mentioned previously, frequency components appearing in the first-order response may appear in higher-order responses also. The function of the option under consideration is to combine these responses and print the response at only the set of distinct frequencies.

The use of the debug option (DB) causes PRANC to print the intermediate results of hybrid analysis of the augmented linear circuit. This option has been incorporated for the debugging of the linear circuit analysis and is not recommended for use during a typical run. An understanding of hybrid analysis [20] is necessary to interpret the output -- which can be quite voluminous -- from the debug run.

4-5. Examples using PRANC

A set of examples are presented in this section to illustrate the use of PRANC for obtaining the steady-state response of nonlinear circuits. Each example will contain the problem statement, the sequence of punched data cards, the computer printed output, and some remarks on the printed output.

Example 4-1: Single Stage Untuned Amplifier Circuit

Consider the untuned, bipolar transistor amplifier of Fig. 4-5. The input source comprises of three frequencies. The sequence of data cards used are shown in Fig. 4-6 and Fig. 4-7. Note that the second card in the sequence, referred to as the option card previously, calls for the pole-residue information (acronym PR) and the printing and plotting of the complete output spectrum (acronym PC) across the 50-ohm resistor present between node 6 and the ground node (card no. 9). By not including AP on the option card, the printing of the responses at all extracted ports was suppressed; instead, only the responses across the 50-ohm resistor and the 0.1 ohm resistor are printed. The transistor parameters used in the example are listed on the computer print-out.

Referring to the computer printed output, we note that all the user-specified information has been listed. A description of the augmented linear network, which is formed after the linear parts of the nonlinear elements have been lumped with the existing linear network, is also listed. In the present examples, six ports were extracted as shown by the port assignment description. The open-circuit impedance matrix is therefore of dimension 6x6. The pole-residue information (see eqn. 4-8) for each of the entries of this matrix is also provided. The transfer function and the output voltage values for the various orders and frequency combinations have also

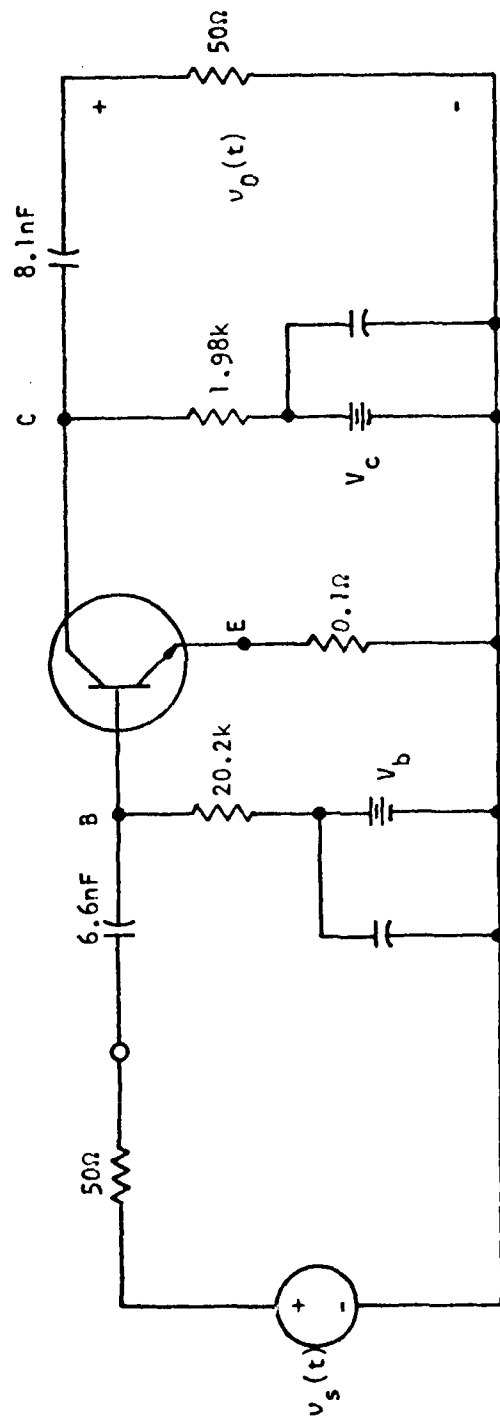


Figure 4-5. A Transistor Amplifier Circuit.

been listed. Finally, the output spectrum across port 3 (node pair 6-0) has been printed and plotted. The total execution time for this example on the CDC 6500 computer is approximately 4.8 seconds.

1.0000E+000.000000003.5000E+06					
1.0000E+000.000000003.0000E+06					
1.0000E+000.000000002.5000E+06					
001000 R 50.00					
25.000E-121.000	0.3400E-090.5900E-0710.100	635.00E+030.0000	1.5000E-12		
4.6	9.27	140.00	0.348	10.000E-03150.00E-030.1250	8.200
007002000TR					
008004000 C4.000E-12					
006006000 R50.00	I#				
005004006 C8.1000E-09					
004004000 R1.9800E+03					
003005000 R0.100	1				
002002000 R20.200E+03					
001001002 C6.6000E-09					
07013 HZ					
PRPC					
EXAMPLE 4-1: SINGLE STAGE UNTUNED AMPLIFIER CIRCUIT					

Figure 4-7. Data Cards for Example 4-1

EXAMPLE 4-1: SINGLE STAGE UNTUNED AMPLIFIER CIRCUIT

USER REQUESTED OPTIONS:

PRINT-OUT: NO
 FREQUENCY SWEEP CAPABILITY: NO
 TWO-INPUT CIRCUITS: NO
 STATE EQUATION PRINT-OUT: NO
 EIGENVALUES/VECA/ MATRIX PRINT-OUT: NO
 OPEN-CIRCUIT IMPEDANCE MATRIX PRINT-OUT: YES
 COMPLETE OUTPUT SPECTRUM PLO: YES
 ALL EXTRACTED PORT OUTPUTS: NO

NETWORK DESCRIPTION:

LINEAR ELEMENTS				
BRANCH NUMBER	FROM NODE	TO NODE	ELEMENT TYPE	CONTROL- VALUE BRANCH
1	1	2	1	5.000E-03
2	2	3	2	3.000E-04
3	3	4	3	1.000E-01
4	4	5	4	1.000E-03
5	5	6	5	3.000E-03
6	6	7	6	3.000E-04
7	7	8	7	4.000E-12

TRANSISTOR PARAMETERS:

IS= 4.000 VDS= 0.270 VDS0=140.000 WJ= .248
 IC= 1.000E-02 ICMAX= 1.500E-01 Q= .185 WFMK= 8.20
 RE= 2.500E-11 RFE= 1.00 QK2= 5.000E-03
 RD= 10.000 QJ= 5.000E-03 QJ2= 1.500E-12

NONLINEAR ELEMENTS

			POLYNOMIAL COEFFICIENTS	
FROM NODE	TO NODE	TYPE	CONTROL (1) (2)	
2	5	NR		A1= 4.000E-01 A2= 3.200E+00 A3= 1.037E+02
4	2	NR	14 13	A10= 3.242E-01 A31= 1.070E-03 A20= 7.345E+00 A23= 3.999E-03 A11= 7.100E-07 A30= 9.314E+01 A03= 3.000E-10 A21= 1.274E-05 A12= 1.555E-07
4	3	NR		A1= 1.131E-11 A2= -2.070E-13 A3= 4.934E-15

SOURCE INFORMATION:

FROM	1	TO	0	IMPEDANCE	5.000E+01	R	PHASE(DEG)
FREQUENCY	VALUE(HZ)	AMPLITUDE					
.....					
1	2.500E+06	1.000E+00					0
2	3.000E+06	1.000E+00					0
3	3.500E+06	1.000E+00					0

AUGMENTED LINEAR NETWORK DESCRIPTION

BRANCH NUMBER	FROM NODE	TO NODE	ELEMENT TYPE	ELEMENT VALUE	CONTROL BRANCH
1	1	2	C	6.600E-09	-0
14	3	5	C	1.014E-09	0
24	4	3	C	1.152E-11	0
13	4	2	C	1.500E-12	0
5	4	6	C	8.100E-09	-0
8	4	0	C	4.000E-12	-0
3	5	4	R	1.000E-01	-0
4	4	0	R	1.980E+03	-0
6	6	0	R	5.000E+01	-0
16	4	3	R	6.350E+05	0
17	1	0	R	5.000E+01	0
7	2	3	R	1.010E+01	0
2	2	0	R	2.020E+04	-0
19	3	5	R	4.332E-01	0
23	4	3	G	3.842E-01	14
22	4	3	UC	1.871E-08	13
18	1	0	I	0	0
9	5	0	I	0	0
10	6	0	I	0	0
11	3	5	I	0	0
12	4	2	I	0	0
21	4	2	I	0	0

PORT ASSIGNMENTS:

PORT NUMBER	FROM	TO
1	1	0
2	5	0
3	6	0
4	3	5
5	4	3
6	4	2

OPEN CIRCUIT IMPEDANCE MATRIX

Z(1, 1): RESIDUE
 -1.83986E+11+J 0
 -7.32280E+11+J 0
 1.17889E+08+J 0
 -6.14623E+08+J 0
 -5.54126E+07+J 0
 6.39202E+00+J 0
 EIGENVALUE
 -3.38784E+12+J 0
 -8.71553E+10+J 0
 -1.51107E+09+J 0
 -5.30504E+07+J 0
 -1.84473E+05+J 0
 -6.16155E+04+J 0

CONSTANT= 0
 Z(1, 2): RESIDUE
 -2.49301E+11+J 0
 -6.99546E+08+J 0
 -1.05010E+08+J 0
 1.18303E+09+J 0
 1.10707E+05+J 0
 -1.12321E+02+J 0
 EIGENVALUE
 -3.38784E+12+J 0
 -8.71553E+10+J 0
 -1.51107E+09+J 0
 -5.30504E+07+J 0
 -1.84473E+05+J 0
 -6.16155E+04+J 0

CONSTANT= 0
 Z(1, 3): RESIDUE
 -1.84519E+11+J 0
 -6.60300E+10+J 0
 5.62370E+08+J 0
 2.42051E+07+J 0
 -2.91769E+04+J 0
 -9.91836E+01+J 0
 EIGENVALUE
 -3.38784E+12+J 0
 -8.71553E+10+J 0
 -1.51107E+09+J 0
 -5.30504E+07+J 0
 -1.84473E+05+J 0
 -6.16155E+04+J 0

CONSTANT= 0
 Z(1, 4): RESIDUE
 7.32682E+02+J 0
 -2.05077E+03+J 0
 1.58734E+08+J 0
 -7.04654E+08+J 0
 2.42354E+07+J 0
 -3.05324E+02+J 0
 EIGENVALUE
 -3.38784E+12+J 0
 -8.71553E+10+J 0
 -1.51107E+09+J 0
 -5.30504E+07+J 0
 -1.84473E+05+J 0
 -6.16155E+04+J 0

CONSTANT= 0
 Z(1, 5): RESIDUE
 6.40496E+10+J 0
 -6.51615E+10+J 0
 4.03767E+08+J 0
 7.85500E+03+J 0
 -2.44263E+07+J 0
 3.45590E+02+J 0
 EIGENVALUE
 -3.38784E+12+J 0
 -8.71553E+10+J 0
 -1.51107E+09+J 0
 -5.30504E+07+J 0
 -1.84473E+05+J 0
 -6.16155E+04+J 0

CONSTANT= 0
 Z(1, 6): RESIDUE
 -5.32377E+02+J 0
 6.65186E+11+J 0
 4.43759E+08+J 0
 6.02600E+03+J 0
 -3.55029E+07+J 0
 3.46727E+02+J 0
 EIGENVALUE
 -3.38784E+12+J 0
 -8.71553E+10+J 0
 -1.51107E+09+J 0
 -5.30504E+07+J 0
 -1.84473E+05+J 0
 -6.16155E+04+J 0

CONSTANT= 0
 2, 1, 1: RESIDUE
 -2.49301E+00
 -3.52014E+00
 1.73427E+00
 -2.50471E+00
 1.01002E+00
 -1.52751E+00
 0 0 0 0 0 0
 EIGENVALUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00

CONSTANT= 0
 2, 2, 2: RESIDUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00
 0 0 0 0 0 0
 EIGENVALUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00

CONSTANT= 0
 2, 3, 3: RESIDUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00
 0 0 0 0 0 0
 EIGENVALUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00

CONSTANT= 0
 2, 4, 4: RESIDUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00
 0 0 0 0 0 0
 EIGENVALUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00

CONSTANT= 0
 2, 5, 5: RESIDUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00
 0 0 0 0 0 0
 EIGENVALUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00

CONSTANT= 0
 2, 6, 6: RESIDUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00
 0 0 0 0 0 0
 EIGENVALUE
 -3.52724E+00
 -3.71533E+00
 -1.51102E+00
 -3.50393E+00
 -1.51432E+00
 -3.15155E+00

CONSTANT= 0
 Z(3, 1): RESIDUE
 -1.04516E+11+J
 -3.50476E+10+J
 -1.22870E+10+J
 1.31063E+10+J
 -4.03303E+03+J
 4.83321E+05+J
 EIGENVALUE
 -3.33784E+12+J
 -3.71553E+10+J
 -1.51107E+03+J
 -5.30504E+07+J
 -1.84473E+03+J
 -5.16155E+04+J

CONSTANT= 0
 Z(3, 2): RESIDUE
 -2.50022E+11+J
 -5.20041E+07+J
 1.08447E+06+J
 -2.5277E+07+J
 9.06345E+05+J
 -3.50307E+02+J
 EIGENVALUE
 -3.33784E+12+J
 -3.71553E+10+J
 -1.51107E+03+J
 -5.30504E+07+J
 -1.84473E+03+J
 -5.16155E+04+J

CONSTANT= 0
 Z(3, 3): RESIDUE
 -1.85052E+11+J
 -5.94003E+08+J
 -5.03104E+10+J
 -5.16167E+03+J
 -2.36344E+03+J
 -7.50104E+04+J
 EIGENVALUE
 -3.33784E+12+J
 -3.71553E+10+J
 -1.51107E+03+J
 -5.30504E+07+J
 -1.84473E+03+J
 -5.16155E+04+J

CONSTANT= 0
 Z(3, 4): RESIDUE
 7.34302E+03+J
 -1.84403E+07+J
 -1.59103E+10+J
 1.50257E+10+J
 1.59316E+03+J
 -2.34344E+07+J
 EIGENVALUE
 -3.33784E+12+J
 -3.71553E+10+J
 -1.51107E+03+J
 -5.30504E+07+J
 -1.84473E+03+J
 -5.16155E+04+J

CONSTANT= 0
 Z(3, 5): RESIDUE
 6.42343E+10+J
 -5.85941E+09+J
 -4.27032E+10+J
 -1.54337E+10+J
 -2.00041E+03+J
 2.64177E+07+J
 EIGENVALUE
 -3.33784E+12+J
 -3.71553E+10+J
 -1.51107E+03+J
 -5.30504E+07+J
 -1.84473E+03+J
 -5.16155E+04+J

CONSTANT= 0
 Z(3, 6): RESIDUE
 -5.33518E+08+J
 5.99045E+10+J
 -4.62553E+10+J
 -1.26302E+10+J
 -2.51573E+03+J
 2.65047E+07+J
 EIGENVALUE
 -3.33784E+12+J
 -3.71553E+10+J
 -1.51107E+03+J
 -5.30504E+07+J
 -1.84473E+03+J
 -5.16155E+04+J

CONSTANT=	0			
Z(4, 1):	RESIDUE		EIGENVALUE	
	7.25520E+08+J	0	-3.28784E+12+J	0
	7.55933E+07+J	0	-8.71553E+10+J	0
	-1.55533E+03+J	0	-1.51107E+09+J	0
	-6.63107E+05+J	0	-5.20504E+07+J	0
	2.34283E+07+J	0	-1.84473E+06+J	0
	-3.11195E+02+J	0	-3.16155E+04+J	0
CONSTANT=	0			
Z(4, 2):	RESIDUE		EIGENVALUE	
	9.83077E+03+J	0	-3.28784E+12+J	0
	7.51755E+04+J	0	-8.71553E+10+J	0
	1.42102E+03+J	0	-1.51107E+09+J	0
	1.28537E+03+J	0	-5.20504E+07+J	0
	-4.63035E+04+J	0	-1.84473E+06+J	0
	5.45333E-01+J	0	-3.16155E+04+J	0
CONSTANT=	0			
Z(4, 3):	RESIDUE		EIGENVALUE	
	7.25520E+08+J	0	-3.28784E+12+J	0
	7.55933E+07+J	0	-8.71553E+10+J	0
	-1.55533E+03+J	0	-1.51107E+09+J	0
	-6.63107E+05+J	0	-5.20504E+07+J	0
	2.34283E+07+J	0	-1.84473E+06+J	0
	-3.11195E+02+J	0	-3.16155E+04+J	0
CONSTANT=	0			
Z(4, 4):	RESIDUE		EIGENVALUE	
	7.25520E+08+J	0	-3.28784E+12+J	0
	7.55933E+07+J	0	-8.71553E+10+J	0
	-1.55533E+03+J	0	-1.51107E+09+J	0
	-6.63107E+05+J	0	-5.20504E+07+J	0
	2.34283E+07+J	0	-1.84473E+06+J	0
	-3.11195E+02+J	0	-3.16155E+04+J	0
CONSTANT=	0			
Z(4, 5):	RESIDUE		EIGENVALUE	
	7.25520E+08+J	0	-3.28784E+12+J	0
	7.55933E+07+J	0	-8.71553E+10+J	0
	-1.55533E+03+J	0	-1.51107E+09+J	0
	-6.63107E+05+J	0	-5.20504E+07+J	0
	2.34283E+07+J	0	-1.84473E+06+J	0
	-3.11195E+02+J	0	-3.16155E+04+J	0

CONSTANT= 0
Z(5, 1): RESIDUE
6.40588E+10+J
-6.52321E+10+J
-1.21250E+10+J
1.31902E+10+J
1.29165E+08+J
-1.90917E+07+J

EIGENVALUE
-3.38784E+12+J
-8.71553E+10+J
-1.51107E+09+J
-5.30504E+07+J
-1.84473E+06+J
-6.16155E+04+J

CONSTANT= 0
Z(5, 2): RESIDUE
8.67967E+10+J
-6.23161E+07+J
1.08004E+08+J
-2.53885E+07+J
-2.58054E+05+J
3.35480E+04+J

EIGENVALUE
-3.38784E+12+J
-8.71553E+10+J
-1.51107E+09+J
-5.30504E+07+J
-1.84473E+06+J
-6.16155E+04+J

CONSTANT= 0
Z(5, 3): RESIDUE
6.42421E+10+J
-5.88538E+09+J
-5.78403E+10+J
-5.19457E+08+J
6.80094E+04+J
2.56242E+06+J

EIGENVALUE
-3.38784E+12+J
-8.71553E+10+J
-1.51107E+09+J
-5.30504E+07+J
-1.84473E+06+J
-6.16155E+04+J

CONSTANT= 0
Z(5, 4): RESIDUE
-2.55091E+08+J
-1.82684E+07+J
-1.57089E+10+J
1.51225E+10+J
-5.67018E+07+J
9.16425E+08+J

EIGENVALUE
-3.38784E+12+J
-8.71553E+10+J
-1.51107E+09+J
-5.30504E+07+J
-1.84473E+06+J
-6.16155E+04+J

CONSTANT= 0
Z(5, 5): RESIDUE
-2.22995E+10+J
-5.80464E+09+J
-4.21449E+10+J
-1.55924E+10+J
5.69258E+07+J
-1.03221E+09+J

EIGENVALUE
-3.38784E+12+J
-8.71553E+10+J
-1.51107E+09+J
-5.30504E+07+J
-1.84473E+06+J
-6.16155E+04+J

CONSTANT= 0
Z(5, 6): RESIDUE
1.85353E+08+J
5.93446E+10+J
-4.55451E+10+J
-1.29322E+10+J
8.22899E+07+J
-1.03531E+09+J

EIGENVALUE
-3.38784E+12+J
-8.71553E+10+J
-1.51107E+09+J
-5.30504E+07+J
-1.84473E+06+J
-6.16155E+04+J

CONSTANT=	0		
Z(6, 1):	RESIDUE	EIGENVALUE	
-5.32521E+08+J	0	-3.38784E+12+J	0
6.85405E+11+J	0	-8.71553E+10+J	0
-1.23246E+10+J	0	-1.51107E+03+J	0
1.30763E+10+J	0	-5.30504E+07+J	0
1.17993E+03+J	0	-1.84473E+05+J	0
-1.90917E+07+J	0	-6.16155E+04+J	0
CONSTANT=	0		
Z(6, 2):	RESIDUE	EIGENVALUE	
-7.21559E+03+J	0	-3.38784E+12+J	0
6.36315E+08+J	0	-8.71553E+10+J	0
1.10316E+03+J	0	-1.51107E+03+J	0
-2.51693E+07+J	0	-5.30504E+07+J	0
-2.35733E+05+J	0	-1.84473E+05+J	0
3.55481E+04+J	0	-6.16155E+04+J	0
CONSTANT=	0		
Z(6, 3):	RESIDUE	EIGENVALUE	
-5.34065E+08+J	0	-3.38784E+12+J	0
6.01245E+10+J	0	-8.71553E+10+J	0
-5.98785E+10+J	0	-1.51107E+03+J	0
-3.14370E+03+J	0	-5.30504E+07+J	0
6.21863E+04+J	0	-1.84473E+05+J	0
2.62243E+05+J	0	-6.16155E+04+J	0
CONSTANT=	0		
Z(6, 4):	RESIDUE	EIGENVALUE	
1.16043E+03+J	0	-3.38784E+12+J	0
1.25534E+03+J	0	-8.71553E+10+J	0
-1.69462E+10+J	0	-1.51107E+03+J	0
1.42316E+10+J	0	-5.30504E+07+J	0
-5.17973E+07+J	0	-1.84473E+05+J	0
9.16426E+07+J	0	-6.16155E+04+J	0
CONSTANT=	0		
Z(6, 5):	RESIDUE	EIGENVALUE	
1.85282E+08+J	0	-3.38784E+12+J	0
5.93000E+10+J	0	-8.71553E+10+J	0
-4.50474E+10+J	0	-1.51107E+03+J	0
-1.54577E+10+J	0	-5.30504E+07+J	0
5.20120E+07+J	0	-1.84473E+05+J	0
-1.03221E+09+J	0	-6.16155E+04+J	0
CONSTANT=	0		
Z(6, 6):	RESIDUE	EIGENVALUE	
-1.54089E+06+J	0	-3.38784E+12+J	0
-6.06262E+11+J	0	-8.71553E+10+J	0
-4.66286E+10+J	0	-1.51107E+03+J	0
-1.28205E+10+J	0	-5.30504E+07+J	0
7.58107E+07+J	0	-1.84473E+05+J	0
-1.03561E+09+J	0	-6.16155E+04+J	0

CONSTANT= 0

FIRST ORDER: FREQUENCY(1) = 2.500E+06 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.00336E-02	-1.32885E-03	1.01212E-02	-3.98954E+01	1.00336E-02	-1.32885E-03	1.01212E-02	-7.54
2	-4.29645E+00	7.73651E-01	4.36555E+00	1.28008E+01	-4.29645E+00	7.73651E-01	4.36555E+00	169.79
3

FIRST ORDER: FREQUENCY(2) = 3.000E+06 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	9.77794E-03	-1.94875E-03	9.97024E-03	-4.00259E+01	9.77794E-03	-1.94875E-03	9.97024E-03	-11.27
2	-4.16119E+00	1.07990E+00	4.29904E+00	1.26674E+01	-4.16119E+00	1.07990E+00	4.29904E+00	165.45
3

FIRST ORDER: FREQUENCY(3) = 3.500E+06 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	9.47540E-03	-2.46325E-03	9.79034E-03	-4.01840E+01	9.47540E-03	-2.46325E-03	9.79034E-03	-14.57
2	-4.00289E+00	1.33561E+00	4.21984E+00	1.25059E+01	-4.00289E+00	1.33561E+00	4.21984E+00	161.55
3

SECOND ORDER: FREQUENCY(1, 1) = 5.000E+06 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	2.05574E-02	1.72855E-03	2.06397E-02	-3.37059E+01	1.02837E-02	8.63346E-04	1.03195E-02	4.80
2	-5.64119E+00	-1.10589E+00	9.70441E+00	1.97394E+01	-4.82059E+00	-5.52945E-01	4.85220E+00	-173.46
3

SECOND ORDER: FREQUENCY(1, 2) = 5.500E+06 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	2.10787E-02	1.72933E-04	2.10794E-02	-3.35228E+01	2.10787E-02	1.72933E-04	2.10794E-02	.47
2	-9.93577E+00	-3.65359E-01	9.94248E+00	1.99499E+01	-9.93577E+00	-3.65359E-01	9.94248E+00	-177.89
3

SECOND ORDER: FREQUENCY(1, 3) = 6.000E+06 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20 LOG MAG	PEAK	IMAGINARY	MAGNITUDE	PHASE DEG
1	2.13517E-02	-1.32955E-03	2.1331E-02	-3.33945E+01	2.13517E-02	-1.32955E-03	2.1331E-02	-3.55
2	2.13517E-02	-1.32955E-03	2.1331E-02	-3.33945E+01	2.13517E-02	-1.32955E-03	2.1331E-02	177.95
3	-1.01112E+01	3.60322E-01	1.01175E+01	2.01019E+01	-1.01112E+01	3.60322E-01	1.01175E+01	
FREQUENCY(1,-1) = 0 HZ								
TRANSFER FUNCTION								
PORT NO	REAL	IMAGINARY	MAGNITUDE	20 LOG MAG	PEAK	IMAGINARY	MAGNITUDE	PHASE DEG
1	3.05822E-03	0	3.05322E-03	-5.02305E+01	3.05822E-03	0	3.05322E-03	0
2	3.05822E-03	0	3.05322E-03	-5.02305E+01	3.05822E-03	0	3.05322E-03	0
3	6.60905E-11	0	6.60905E-11	-2.03597E+02	6.60905E-11	0	6.60905E-11	0
FREQUENCY(2, 2) = 5.000E+05 HZ								
TRANSFER FUNCTION								
PORT NO	REAL	IMAGINARY	MAGNITUDE	20 LOG MAG	PEAK	IMAGINARY	MAGNITUDE	PHASE DEG
1	2.14333E-02	-1.32955E-03	2.14327E-02	-3.33853E+01	2.14333E-02	-1.32955E-03	2.14327E-02	-3.53
2	2.14333E-02	-1.32955E-03	2.14327E-02	-3.33853E+01	2.14333E-02	-1.32955E-03	2.14327E-02	177.53
3	-1.02240E+01	1.02934E+00	1.02934E+01	2.01297E+01	-1.02240E+01	1.02934E+00	1.02934E+01	
FREQUENCY(2, 2) = 5.000E+05 HZ								
TRANSFER FUNCTION								
PORT NO	REAL	IMAGINARY	MAGNITUDE	20 LOG MAG	PEAK	IMAGINARY	MAGNITUDE	PHASE DEG
1	2.14333E-02	-1.32955E-03	2.14327E-02	-3.33853E+01	2.14333E-02	-1.32955E-03	2.14327E-02	-3.53
2	2.14333E-02	-1.32955E-03	2.14327E-02	-3.33853E+01	2.14333E-02	-1.32955E-03	2.14327E-02	177.53
3	-1.02240E+01	1.02934E+00	1.02934E+01	2.01297E+01	-1.02240E+01	1.02934E+00	1.02934E+01	
FREQUENCY(2,-1) = 5.000E+05 HZ								
TRANSFER FUNCTION								
PORT NO	REAL	IMAGINARY	MAGNITUDE	20 LOG MAG	PEAK	IMAGINARY	MAGNITUDE	PHASE DEG
1	1.14377E-02	4.32549E-03	1.25032E-02	-3.80560E+01	1.14377E-02	4.32549E-03	1.25032E-02	23.19
2	1.14377E-02	4.32549E-03	1.25032E-02	-3.80560E+01	1.14377E-02	4.32549E-03	1.25032E-02	23.19
3	-5.11863E+00	-2.30334E+00	5.61300E+00	1.49839E+01	-5.11863E+00	-2.30334E+00	5.61300E+00	-155.77
FREQUENCY(2,-2) = 0 HZ								
TRANSFER FUNCTION								
PORT NO	REAL	IMAGINARY	MAGNITUDE	20 LOG MAG	PEAK	IMAGINARY	MAGNITUDE	PHASE DEG
1	1.14377E-02	4.32549E-03	1.25032E-02	-3.80560E+01	1.14377E-02	4.32549E-03	1.25032E-02	23.19
2	1.14377E-02	4.32549E-03	1.25032E-02	-3.80560E+01	1.14377E-02	4.32549E-03	1.25032E-02	23.19
3	-5.11863E+00	-2.30334E+00	5.61300E+00	1.49839E+01	-5.11863E+00	-2.30334E+00	5.61300E+00	-155.77
FREQUENCY(2,-2) = 0 HZ								
TRANSFER FUNCTION								
PORT NO	REAL	IMAGINARY	MAGNITUDE	20 LOG MAG	PEAK	IMAGINARY	MAGNITUDE	PHASE DEG
1	1.14377E-02	4.32549E-03	1.25032E-02	-3.80560E+01	1.14377E-02	4.32549E-03	1.25032E-02	23.19
2	1.14377E-02	4.32549E-03	1.25032E-02	-3.80560E+01	1.14377E-02	4.32549E-03	1.25032E-02	23.19
3	-5.11863E+00	-2.30334E+00	5.61300E+00	1.49839E+01	-5.11863E+00	-2.30334E+00	5.61300E+00	-155.77

2 2.96590E-03 0 2.96590E-03 -5.0556E+01 2.96590E-03 0 2.96590E-03 0
 3 6.40957E-11 0 6.40957E-11 -2.03863E+02 6.40957E-11 0 6.40957E-11 0
 SECOND ORDER:

FREQUENCY(3, 2) = 7.000E+06 HZ
 TRANSFER FUNCTION
 PORT NO REAL IMAGINARY MAGNITUDE 20LOG MAG

 2 2.14091E-02 -4.88217E-03 2.19151E-02 -3.31851E+01
 3 -1.02152E+01 1.99344E+00 1.04085E+01 2.03478E+01
 SECOND ORDER: FREQUENCY(3, -1) = 1.000E+05 HZ

OUTPUT VOLTAGE
 REAL IMAGINARY MAGNITUDE PHASE DEG

 1.07045E-02 -2.34109E-03 1.09375E-02 -12.34
 -5.10791E+00 9.55720E-01 5.20424E+00 163.96
 OUTPUT VOLTAGE
 REAL IMAGINARY MAGNITUDE PHASE DEG

 1.25117E-02 3.06614E-03 1.26532E-02 12.79
 -6.04523E+00 -1.51459E+00 6.25555E+00 -155.94
 SECOND ORDER: FREQUENCY(3, -2) = 5.000E+05 HZ

TRANSFER FUNCTION
 PORT NO REAL IMAGINARY MAGNITUDE 20LOG MAG

 2 1.53117E-02 3.05514E-03 1.53552E-02 -3.71572E+01
 3 -1.51452E+00 6.23593E+00 1.55321E+01
 SECOND ORDER: FREQUENCY(3, -3) = 0 HZ

OUTPUT VOLTAGE
 REAL IMAGINARY MAGNITUDE PHASE DEG

 1.10768E-02 4.84408E-03 1.20916E-02 23.62
 -4.59166E+00 -2.28341E+00 5.42566E+00 -155.35
 SECOND ORDER: FREQUENCY(3, -3) = 0 HZ

TRANSFER FUNCTION
 PORT NO REAL IMAGINARY MAGNITUDE 20LOG MAG

 2 2.05105E-02 4.24405E-03 1.20915E-02 -3.33503E+01
 3 -2.28341E+00 5.42566E+00 1.43333E+01
 SECOND ORDER: FREQUENCY(1, 1, 1) = 7.000E+06 HZ

OUTPUT VOLTAGE
 REAL IMAGINARY MAGNITUDE PHASE DEG

 2.05105E-02 4.24405E-03 1.20915E-02 23.62
 -4.59166E+00 -2.28341E+00 5.42566E+00 -155.35
 SECOND ORDER: FREQUENCY(1, 1, 2) = 3.000E+05 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	3.53413E-02	-5.67217E-02	6.53303E-02	-2.35005E+01	2.65050E-02	-4.25413E-02	5.01231E-02	-58.07
2	-1.73339E+01	2.58435E+01	3.18035E+01	3.00300E+01	-1.30274E+01	1.99865E+01	2.38541E+01	123.10
3

THIRD ORDER:

FREQUENCY(1, 1, 3) = 8.500E+05 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	2.83151E-02	-5.95123E-02	6.59937E-02	-2.36097E+01	2.12354E-02	-4.47055E-02	4.94968E-02	-64.59
2	-1.40192E+01	2.81506E+01	3.14435E+01	2.99320E+01	-1.05144E+01	2.11132E+01	2.35834E+01	116.47
3

THIRD ORDER:

FREQUENCY(1, 1, -1) = 2.500E+05 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	5.95255E-02	5.68922E-03	6.83510E-02	-2.32405E+01	5.14593E-02	4.26391E-03	5.16458E-02	4.74
2	-0.12634E+01	-3.70764E+00	3.15288E+01	2.99736E+01	-2.34733E+01	-2.84073E+00	2.36451E+01	-173.10
3

THIRD ORDER:

FREQUENCY(1, 1, -2) = 2.000E+05 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	4.53235E-02	1.57547E-02	4.65594E-02	-2.66380E+01	3.28675E-02	1.18150E-02	3.49271E-02	19.77
2	-1.50113E+01	-7.83168E+00	2.11339E+01	2.65008E+01	-1.47053E+01	-5.91125E+00	1.53527E+01	-153.11
3

THIRD ORDER:

FREQUENCY(1, 1, -3) = 1.500E+05 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	5.64431E-02	1.44652E-02	3.91874E-02	-2.81371E+01	2.73222E-02	1.08031E-02	2.93906E-02	21.57
2	-1.61755E+01	-7.02880E+00	1.75367E+01	2.49284E+01	-1.21317E+01	-5.27160E+00	1.32276E+01	-156.51
3

THIRD ORDER:

FREQUENCY(1, 2, 2) = 8.500E+05 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2
3

THIRD ORDER:

FREQUENCY(1, 2, 2) = 8.500E+05 HZ

.....
 2 2.79005E-02 -6.01255E-02 6.52845E-02 -2.35718E+01 -2.09254E-02 4.97134E-02 -65.11
 3 -1.38261E+01 2.83994E+01 3.15861E+01 2.99899E+01 -1.03895E+01 2.12995E+01 2.36896E+01 115.56

THIRD ORDER:

FREQUENCY(1, 2, 3) = 9.000E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	2.05105E-02	-6.20096E-02	6.53452E-02	-2.36957E+01	3.09158E-02	-9.30145E-02	9.80177E-02	-71.61
3	-1.03270E+01	2.94131E+01	3.11733E+01	2.98757E+01	-1.54905E+01	4.41195E+01	4.67559E+01	109.35

THIRD ORDER:

FREQUENCY(1, 2, -1) = 3.000E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	5.21545E-02	-7.10093E-03	5.25589E-02	-2.40742E+01	3.32319E-02	-1.06514E-02	9.38334E-02	-6.52
3	-2.87203E+01	2.16909E+00	2.88104E+01	2.91910E+01	-4.30929E+01	3.25363E+00	4.32156E+01	175.68

THIRD ORDER:

FREQUENCY(1, 2, -2) = 2.500E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	5.55395E-02	9.13512E-03	5.62844E-02	-2.49785E+01	8.34892E-02	1.37027E-02	8.45785E-02	9.32
3	-2.92315E+01	-5.15248E+00	2.98019E+01	2.82329E+01	-3.79228E+01	-7.72873E+00	3.87023E+01	-168.43

THIRD ORDER:

FREQUENCY(1, 2, -3) = 2.000E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	3.50423E-02	1.02317E-02	4.13823E-02	-2.75374E+01	5.97694E-02	1.92775E-02	6.29834E-02	18.40
3	-1.70102E+01	-5.60235E+00	1.95035E+01	2.55933E+01	-2.67803E+01	-1.00243E+01	2.85762E+01	-155.45

THIRD ORDER:

FREQUENCY(1, 3, 3) = 9.500E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	1.03893E-02	-5.80135E-02	5.43275E-02	-2.33320E+01	9.54593E-03	-4.72093E-02	4.82457E-02	-78.10
3	-3.70011E+00	2.60331E+01	3.07172E+01	2.97478E+01	-5.08738E+00	2.24693E+01	2.30383E+01	102.76

THIRD ORDER:

FREQUENCY(1, 3, -1) = 3.500E+05 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	6.02404E-02	-1.03795E-02	6.11875E-02	-2.42557E+01	9.03785E-02	-1.60192E-02	9.17814E-02	-10.05
2	-2.80690E+01	3.87244E+00	2.85533E+01	2.90330E+01	-4.21346E+01	5.81766E+00	4.25344E+01	172.14
3								

THIRD ORDER: FREQUENCY(1, 3, -2) = 3.000E+03 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	4.50593E-02	-2.03728E-03	4.51040E-02	-2.61777E+01	7.35895E-02	-3.13106E-03	7.25592E-02	-2.44
2	-2.25337E+01	7.90857E-02	2.25333E+01	2.70798E+01	-3.38505E+01	1.18628E-01	3.38907E+01	179.80
3								

THIRD ORDER: FREQUENCY(1, 3, -3) = 2.500E+05 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	5.00333E-02	7.29330E-03	5.13474E-02	-2.57885E+01	7.06388E-02	1.09497E-02	7.70211E-02	8.17
2	-2.27011E+01	-4.20552E+00	2.34315E+01	2.74182E+01	-3.45508E+01	-6.24933E+00	3.52373E+01	-169.62
3								

THIRD ORDER: FREQUENCY(2, 2, 2) = 9.000E+05 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	2.10455E-02	-3.10311E-02	3.53133E-02	-2.35894E+01	5.03394E-02	-1.55134E-02	1.54043E-02	-72.12
2	2.10455E-02	3.10311E-02	3.53133E-02	2.35894E+01	-2.45593E+00	7.40712E+00	7.92533E+00	103.22
3								

THIRD ORDER: FREQUENCY(2, 2, 2) = 5.500E+05 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	1.20073E-02	-3.20013E-02	3.43333E-02	-2.37912E+01	5.56045E-02	-6.74888E-02	4.84397E-02	-73.82
2	-2.50673E+01	3.61203E+01	3.62111E+01	2.97635E+01	-4.51175E+00	2.28043E+01	2.31208E+01	103.24
3								

THIRD ORDER: FREQUENCY(2, 2, -1) = 3.500E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1								
2								
3								

THIRD ORDER:
 2 5.37348E-02 -1.98713E-02 5.72914E-02 -2.48382E+01 4.03011E-02 -1.49035E-02 4.29685E-02 -20.29
 3 -2.51982E+01 8.25516E+00 2.65160E+01 2.84702E+01 -1.88987E+01 6.19137E+00 1.98870E+01 151.86

THIRD ORDER: FREQUENCY(2, 2, -2) = 3.000E+06 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	6.75850E-02	9.47438E-04	6.75917E-02	-2.34021E+01	5.06888E-02	7.10579E-04	5.06937E-02	.80
3	-3.11214E+01	-1.63671E+00	3.11644E+01	2.98732E+01	-2.33411E+01	-1.22753E+00	2.33733E+01	-176.99

THIRD ORDER: FREQUENCY(2, 2, -3) = 2.500E+06 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	4.33241E-02	1.20875E-02	4.49787E-02	-2.69339E+01	3.24930E-02	9.06564E-03	3.27340E-02	15.59
3	-1.95821E+01	-6.30260E+00	2.05713E+01	2.62653E+01	-1.46866E+01	-4.72695E+00	1.54265E+01	-162.16

THIRD ORDER: FREQUENCY(2, 3, 3) = 1.000E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	5.44814E-03	-6.32511E-02	6.34853E-02	-2.39455E+01	4.08610E-03	-4.74383E-02	4.76140E-02	-85.03
3	-3.00544E+00	3.01919E+01	3.03411E+01	2.96406E+01	-2.25408E+00	2.26439E+01	2.27553E+01	95.63

THIRD ORDER: FREQUENCY(2, 3, -1) = 4.000E+06 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	5.05720E-02	-2.32806E-02	5.56733E-02	-2.50871E+01	7.58580E-02	-3.49203E-02	8.35093E-02	-24.72
3	-2.39143E+01	9.96256E+00	2.59065E+01	2.82682E+01	-3.58715E+01	1.49438E+01	3.88597E+01	157.33

THIRD ORDER: FREQUENCY(2, 3, -2) = 3.500E+06 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	6.00695E-02	-1.10751E-02	6.10823E-02	-2.42817E+01	9.01048E-02	-1.66126E-02	9.16234E-02	-10.45
3	-2.80145E+01	4.06843E+00	2.83085E+01	2.90383E+01	-4.20219E+01	5.10272E+00	4.24627E+01	171.74

THIRD ORDER: FREQUENCY(2, 3, -3) = 3.000E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	5.44314E-02	5.25902E-03	5.42143E-02	-2.52380E+01	8.16922E-02	7.89413E-03	8.20717E-02	5.51
2	-2.42353E+01	-3.40132E+00	2.52156E+01	2.80335E+01	-3.74760E+01	-5.10193E+00	3.78237E+01	-172.25
3								

THIRD ORDER: FREQUENCY(3, 3, 3) = 1.050E+07 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.62324E-03	-6.22025E-02	5.23251E-02	-2.41057E+01	-4.09055E-04	-1.55759E-02	1.55812E-02	-91.50
2	4.55065E-01	2.53035E+01	2.98028E+01	2.94889E+01	1.08766E-01	7.45140E+00	7.45215E+00	89.16
3								

THIRD ORDER: FREQUENCY(3, 3, -1) = 4.500E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	4.68035E-02	-2.51434E-02	5.37929E-02	-2.50335E+01	3.51288E-02	-1.89409E-02	4.03447E-02	-29.46
2	-2.10159E+01	1.15378E+01	2.51474E+01	2.80099E+01	-1.67388E+01	3.69086E+00	1.89305E+01	152.56
3								

THIRD ORDER: FREQUENCY(3, 3, -2) = 4.000E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	5.07245E-02	-2.23935E-02	5.55335E-02	-2.50901E+01	3.80434E-02	-1.71745E-02	4.17404E-02	-24.30
2	-2.33301E+01	9.72542E+00	2.59002E+01	2.82551E+01	-1.79551E+01	7.33983E+00	1.94251E+01	157.80
3								

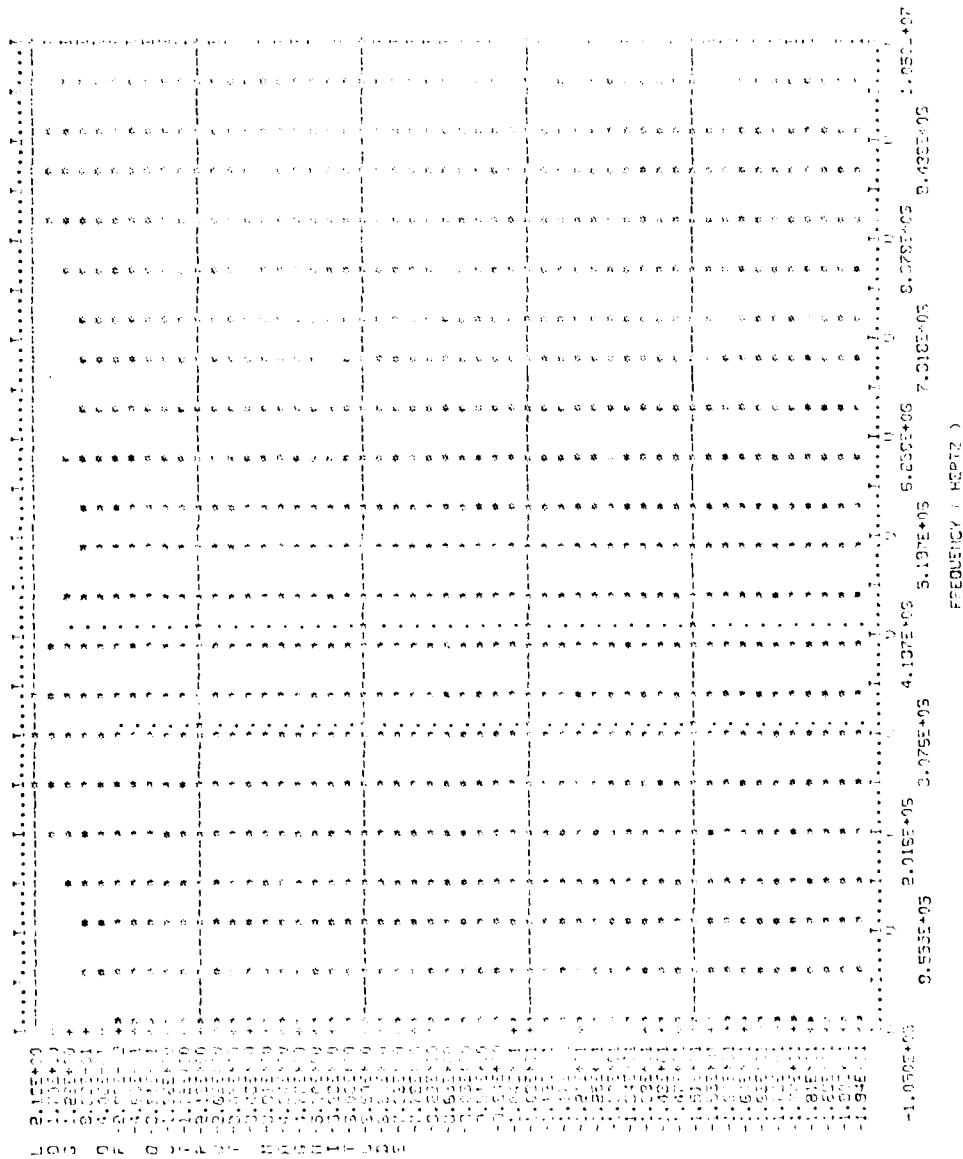
THIRD ORDER: FREQUENCY(3, 3, -3) = 3.500E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	6.55880E-02	-3.13439E-03	6.55722E-02	-2.36553E+01	4.91885E-02	-2.35099E-03	4.92546E-02	-2.74
2	-3.04699E+01	2.92175E-01	3.04713E+01	2.96778E+01	-2.28524E+01	2.19131E-01	2.28535E+01	179.45
3								

SINUSOIDAL STEADY-STATE OUTPUT RESPONSE AT PORT 3

FREQUENCY HERTZ	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2.50E+06	-1.064E+02	-2.242E+01	1.088E+02	-1.681E+02
3.00E+06	-1.336E+02	-4.037E+00	1.337E+02	-1.783E+02
3.50E+06	-1.219E+02	1.700E+01	1.231E+02	1.721E+02
5.00E+06	-4.821E+00	-5.529E-01	4.852E+00	-1.735E+02
5.50E+06	-9.936E+00	-3.654E-01	9.942E+00	-1.779E+02
6.00E+06	-1.518E+01	5.789E-01	1.519E+01	1.778E+02
0	1.919E-10	0	1.919E-10	0
6.50E+06	-1.022E+01	1.204E+00	1.029E+01	1.733E+02
5.00E+05	-1.005E+01	-4.567E+00	1.104E+01	-1.556E+02
7.00E+06	-5.108E+00	9.967E-01	5.204E+00	1.690E+02
1.00E+06	-6.049E+00	-1.515E+00	6.236E+00	-1.659E+02
7.50E+06	-5.162E+00	6.098E+00	7.989E+00	1.302E+02
8.00E+06	-1.303E+01	1.998E+01	2.385E+01	1.231E+02
8.50E+06	-2.088E+01	4.241E+01	4.728E+01	1.162E+02
2.00E+06	-4.147E+01	-1.594E+01	4.443E+01	-1.590E+02
1.50E+06	-1.213E+01	-5.272E+00	1.323E+01	-1.565E+02
9.00E+06	-1.802E+01	5.153E+01	5.459E+01	1.093E+02
9.50E+06	-9.992E+00	4.507E+01	4.617E+01	1.025E+02
1.00E+07	-2.254E+00	2.264E+01	2.276E+01	9.593E+01
4.00E+06	-5.386E+01	2.228E+01	5.828E+01	1.575E+02
1.05E+07	1.088E-01	7.451E+00	7.452E+00	8.916E+01
4.50E+06	-1.674E+01	8.691E+00	1.886E+01	1.526E+02

RESPONSE MAGNITUDE VS FREQUENCY



TIME FOR FORMING ZOC(SEC)	1.1530	
TIME FOR OBTAINING OUTPUT SPECTRUM(SEC)	3.6050	
TOTAL EXECUTION TIME(SEC)	4.7580	

*** P R A N C ***
SEPTEMBER 1979 VERSION

Example 4-2: Two-Stage Tuned Amplifier Circuit

Consider the two-stage tuned amplifier circuit of Fig. 4-8. The input source comprises of two frequencies:

$$v_s(t) = \cos(2\pi 3 \times 10^6 t) + \cos(2\pi 3.25 \times 10^6 t)$$

The sequence of data cards used are shown in Fig. 4-9. In this example the frequency sweep capability (FS on the option card) offered by PRANC was used.

The computer printed output is similar to that for Example 4-1. The two transistors in the circuit account for six nonlinear elements. Altogether nine ports were extracted for the Volterra series analysis, two of which were the desired output ports.

The maximum number of frequency increments specified were five. Note that, as the frequency sweep is implemented, the set of input frequencies are printed before the transfer function and output voltage values. Considering the execution times, we note that the formation of the 9x9 open-circuit matrix took approximately 4 seconds on the CDC 6500 computer; the calculation and the printing of the transfer functions and output voltage values at the positive frequency values (approximately 90 points*) required approximately 18 seconds. The entire program execution required less than 22 seconds.

*The actual number of points is approximately 150, since transfer functions at negative frequencies are required in the calculations.

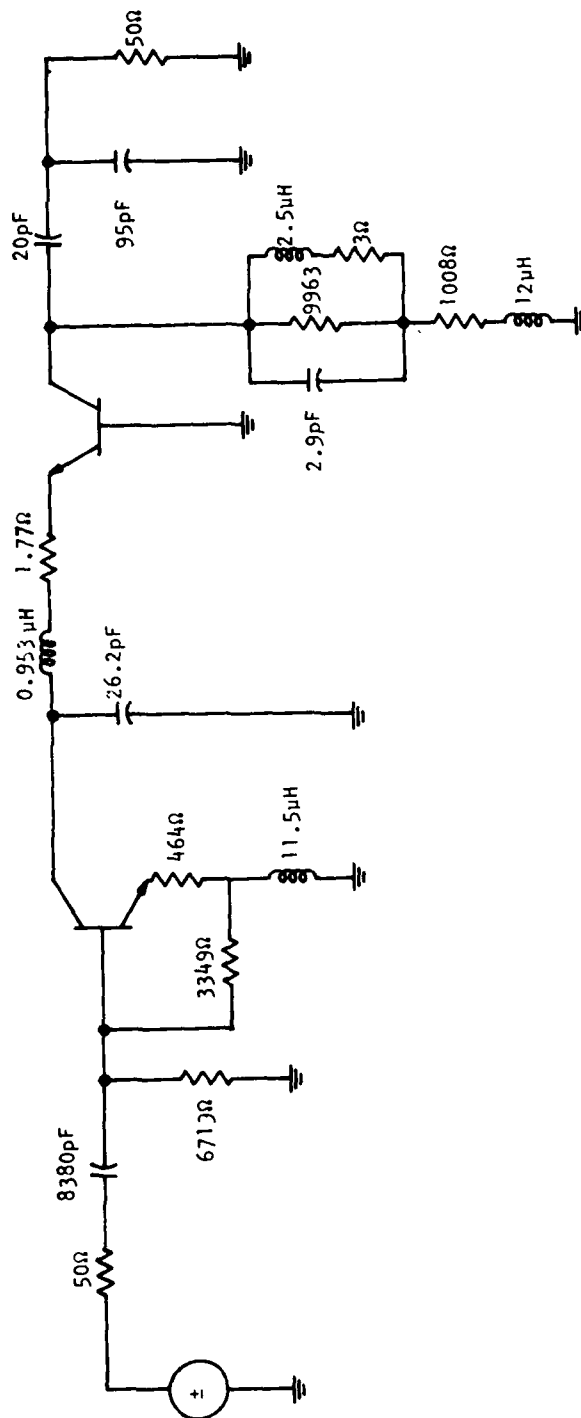


Figure 4-8. Circuit Diagram for Example 4-2

1.000	0.0000	3.2500E+066.3250E+0704	
1.000	0.0000	3.0000E+066.3000E+0704	
004000	R50.000		
1.4000E-121.000	25.000E-127.5000E-0990.000	2.0000E+05 0.000	0.000
4.5	9.600	50.000 0.086	3.7000E-0320.000E-030.300
010000	TR		51.40
2.4500E-111.0000	3.4000E-105.9100E-0810.1000	6.3500E+050.000	1.5000E-12
4.6	9.30000	140.0000000.348000003.7000E-03150.00E-030.1250000008.200	
004005	TR		
019012000	L12.000E-06		
018011012	R1.0080E+03		
017014000	R50.000	1	
016014000	C95.000E-12		
015002014	C20.000E-12		
014013011	R3.0000		
013002013	L2.5700E-06		
012002011	R9.9630E+03		
011002011	C2.9000E-12		
009010003	R1.770		
008007010	L0.9530E-06		
007007000	C26.200E-12	1	
006009000	L11.500E-06		
005008009	R4.6400E+02		
003005009	R3.3490E+03		
002005000	R6.7190E+03		
001004005	C8.3800E-09		
17022	H2LIN		
FS			
EXAMPLE 4-2: TWO-STAGE TUNED AMPLIFIER CIRCUIT			

Figure 4-9. Data Cards for Example 4-2

EXAMPLE 4-2: TWO-STAGE TUNED AMPLIFIER CIRCUIT

USER REQUESTED OPTIONS:
 DEBUG PRINT-OUT: NO
 FREQUENCY SWEEP CAPABILITY: YES
 TWO-INPUT CIRCUIT: NO
 STATE EQUATION PRINT-OUT: NO
 EIGENVALUES MODAL MATRIX PRINT-OUT: NO
 OPEN-CIRCUIT IMPEDANCE MATRIX PRINT-OUT: NO
 COMPLETE OUTPUT SPECTRUM PLOT: NO
 ALL EXTRACTED PORT OUTPUTS: NO

NETWORK DESCRIPTION:

LINEAR ELEMENTS

BRANCH NUMBER	FROM NODE	TO NODE	ELEMENT TYPE	ELEMENT VALUE	CONTROL BRANCH
1	4	5	C	8.380E-09
2	5	0	R	6.715E+03
3	5	9	R	3.349E+03
5	8	9	R	4.640E+02
6	9	0	L	1.50E-05
7	7	0	C	2.620E-11
8	7	10	L	9.530E-07
9	10	3	R	1.770E+00
11	2	11	C	2.900E-12
12	2	11	R	9.863E+03
13	2	13	L	2.570E-06
14	13	11	R	3.000E+00
15	2	14	C	2.000E-11
16	14	0	C	9.500E-11
18	11	12	R	1.008E+03
17	12	0	L	1.200E-05
19	14	0	R	5.000E+01

TRANSISTOR PARAMETERS:

N= 4.600 UCB= 9.300 UCB0=140.000 MU= .348
 IC= 3.700E-03 ICMAX= 1.500E-01 A= .125 HFEMAX= 8.20
 K= 2.450E-11 REF= 1.00 CJE= 3.400E-10 C#2= 5.910E-08
 RB= 10.100 RC= 6.350E+05 C1= 0 C3= 1.500E-12

TRANSISTOR PARAMETERS:

N= 4.500 UCB= 9.600 UCB0= 50.000 MU= .086
 IC= 3.700E-03 ICMAX= 2.000E-02 A= .300 HFEMAX= 51.40
 K= 1.400E-12 REF= 1.00 CJE= 2.500E-11 C#2= 7.500E-09
 RB= 90.000 RC= 2.000E+05 C1= 0 C3= 0

NONLINEAR ELEMENTS

FROM NODE	TO NODE	TYPE	CONTROL (1)	CONTROL (2)	POLYNOMIAL COEFFICIENTS					
6	8	NR			A1=	1.6286E-01	A2=	3.0862E+00	A3=	3.8988E+01
7	6	ND	25	24	A10=	1.4285E-01	A01=	7.0034E-09	A20=	2.7446E+00
					A02=	1.3555E-09	A11=	2.7039E-07	A30=	3.4980E+01
					A03=	1.2632E-10	A21=	5.1950E-06	A12=	5.2334E-08
7	6	NC			A1=	1.1275E-11	A2=	-2.0207E-13	A3=	4.8284E-15
1	3	NR			A1=	1.4340E-01	A2=	2.7174E+00	A3=	3.4330E+01
2	1	ND	31	30	A10=	1.4083E-01	A01=	1.0334E-06	A20=	2.6729E+00
					A02=	1.8868E-07	A11=	3.9331E-05	A30=	3.3741E+01
					A03=	1.6457E-08	A21=	7.4643E-04	A12=	7.1807E-06
2	1	NC			A1=	1.1525E-12	A2=	-2.0009E-14	A3=	4.6316E-16

SOURCE INFORMATION:

FROM	TO	0	IMPEDANCE	5.000E+01	R
FREQUENCY	VALUE (HZ)		AMPLITUDE	PHASE (DEG)	
1	3.000E+06		1.000E+00	0	
2	3.250E+06		1.000E+00	0	

FREQUENCY SHEEP TYPE: LIN

MAXIMUM NUMBER OF INCREMENTS= 4

AUGMENTED LINEAR NETWORK DESCRIPTION

BRANCH NUMBER	FROM NODE	TO NODE	ELEMENT TYPE	ELEMENT VALUE	CONTROL BRANCH
1	4	5	C	8.380E-09	-0
25	6	8	C	5.940E-10	0
47	2	1	C	1.153E-12	0
15	14	14	C	2.000E-11	-0
16	14	0	C	9.500E-11	-0
11	2	11	C	2.900E-12	-0
7	2	0	C	2.620E-11	-0
31	1	3	C	5.338E-11	0
24	7	5	C	1.500E-12	0
41	7	6	C	1.128E-11	0
9	10	1	R	1.770E+00	-0
10	0	1	R	9.000E+01	0
5	8	9	R	4.640E+02	-0
12	2	11	R	9.963E+03	-0

4	3	5	6	R	1.010E+01	0
33	2	5	9	R	3.349E+03	-0
19	14	5	1	R	2.000E+05	-0
42	11	14	0	R	6.713E+03	-0
18	11	1	3	R	5.000E+01	-0
27	7	12	12	G	1.434E-01	-0
34	4	6	6	R	1.008E+03	0
36	6	4	0	R	6.350E+05	0
14	6	8	0	R	5.000E+01	0
30	13	11	11	G	1.629E-01	0
17	12	0	0	R	3.000E+00	-0
6	9	0	0	L	1.000E+06	-0
8	7	10	0	L	1.200E-05	-0
13	2	13	10	L	1.150E-05	-0
40	7	13	13	L	9.530E-07	-0
39	7	6	6	UC	2.570E-06	-0
45	7	6	6	UC	1.428E-01	25
35	4	1	1	UC	7.003E-09	24
21	14	0	1	UC	1.033E-06	30
20	7	0	0	I	1.408E-01	31
22	6	8	0	I	0	0
38	7	8	0	I	0	0
28	1	3	0	I	0	0
29	2	1	0	I	0	0
44	2	1	0	I	0	0

PORT ASSIGNMENTS:

PORT NUMBER	MODE	PAIR FROM TO
1
2	4	0
3	14	0
4	7	0
5	6	8
6	7	6
7	1	3
8	2	1
9	2	0

FIRST ORDER:

FREQUENCY(1)= 3.000E+06 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
.....
2	-2.77042E-02	-1.97449E-02	3.40204E-02	-2.93552E+01	-2.77042E-02	-1.97449E-02	3.40204E-02	-144.52
3	-3.77988E-02	-1.54311E-02	4.08273E-02	-2.77810E+01	-3.77988E-02	-1.54311E-02	4.08273E-02	-157.79

FIRST ORDER: FREQUENCY(2) = 3.250E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-3.16702E-02	-1.91258E-02	3.69972E-02	-2.86365E+01	-3.16702E-02	-1.91258E-02	3.69972E-02	-143.87
3	-4.04370E-02	-1.56788E-02	4.34163E-02	-2.72468E+01	-4.04370E-02	-1.56788E-02	4.34163E-02	-158.83

SECOND ORDER: FREQUENCY(1, 1) = 6.000E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.54432E-03	-1.73550E-03	2.32312E-03	-5.25785E+01	-7.72160E-04	-8.67748E-04	1.16153E-03	-131.65
3	-8.01721E-05	-4.81972E-03	4.82039E-03	-4.63384E+01	-4.00861E-05	-2.40986E-03	2.41020E-03	-99.95

SECOND ORDER: FREQUENCY(1, 2) = 6.250E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.80850E-03	-1.72220E-03	2.49740E-03	-5.20502E+01	-1.80850E-03	-1.72220E-03	2.49740E-03	-133.40
3	-3.54335E-04	-4.95729E-03	4.95934E-03	-4.60730E+01	-3.54335E-04	-4.95729E-03	4.95934E-03	-91.09

SECOND ORDER: FREQUENCY(1, -1) = 0 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-2.23976E-13	0	2.23976E-13	-2.52996E+02	-2.23976E-13	0	2.23976E-13	180.00
3	-2.63862E-03	0	2.63862E-03	-5.15725E+01	2.63862E-03	0	2.63862E-03	0

SECOND ORDER: FREQUENCY(2, 2) = 6.500E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-2.08395E-03	-1.67593E-03	2.67425E-03	-5.14560E+01	-1.04198E-03	-8.37965E-04	1.33712E-03	-141.19
3	-6.43542E-04	-5.08384E-03	5.12441E-03	-4.58071E+01	-3.21771E-04	-2.54192E-03	2.58220E-03	-97.21

SECOND ORDER: FREQUENCY(2, -1) = 2.500E+05 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-2.08395E-03	-1.67593E-03	2.67425E-03	-5.14560E+01	-1.04198E-03	-8.37965E-04	1.33712E-03	-141.19
3	-6.43542E-04	-5.08384E-03	5.12441E-03	-4.58071E+01	-3.21771E-04	-2.54192E-03	2.58220E-03	-97.21

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.02843E-06	-1.54661E-05	1.55002E-05	-9.61932E+01	1.02843E-06	-1.54661E-05	1.55002E-05	-85.20
2	2.73984E-03	-1.55166E-04	2.74423E-03	-5.12316E+01	2.73984E-03	-1.55166E-04	2.74423E-03	-3.24
3								

SECOND ORDER:
FREQUENCY(2,-2) = 0 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-2.15878E-13	0	2.15878E-13	-2.5316E+02	-2.15878E-13	0	2.15878E-13	180.00
2	2.63205E-03	0	2.63205E-03	-5.15941E+01	2.63205E-03	0	2.63205E-03	0
3								

THIRD ORDER:
FREQUENCY(1, 1, 1) = 9.000E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.39034E-03	3.61338E-04	1.43652E-03	-5.6837E+01	-3.47584E-04	9.03344E-05	3.59131E-04	163.43
2	-1.51580E-03	-1.62911E-03	2.22523E-03	-5.30525E+01	-3.78950E-04	-4.07277E-04	5.56307E-04	-132.94
3								

THIRD ORDER:
FREQUENCY(1, 1, 2) = 9.250E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.38397E-03	4.96255E-04	1.47496E-03	-5.66244E+01	-1.04173E-03	3.72191E-04	1.10622E-03	180.34
2	-1.63825E-03	-1.58553E-03	2.27988E-03	-5.28418E+01	-1.22869E-03	-1.18915E-03	1.70550E-03	-135.94
3								

THIRD ORDER:
FREQUENCY(1, 1,-1) = 3.000E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	4.91686E-05	-1.49523E-04	1.57400E-04	-7.60599E+01	3.68765E-05	-1.12142E-04	1.18050E-04	-71.80
2	6.51691E-04	-6.78205E-04	9.40565E-04	-6.05322E+01	4.88769E-04	-5.08654E-04	7.05426E-04	-45.14
3								

THIRD ORDER:
FREQUENCY(1, 1,-2) = 2.750E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	4.92228E-05	-1.24330E-04	1.33670E-04	-7.74793E+01	3.62421E-05	-9.34723E-05	1.00253E-04	-63.81
2	6.55304E-04	-6.37733E-04	9.37236E-04	-6.05630E+01	5.15103E-04	-4.78304E-04	7.02927E-04	-42.63
3								

THIRD ORDER: FREQUENCY(1, 2, 2) = 9.500E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.37264E-03	6.31730E-04	1.51103E-03	-5.84145E+01	-1.02948E-03	4.73797E-04	1.13327E-03	153.29
2	-1.76050E-03	-1.53393E-03	2.33501E-03	-5.26342E+01	-1.32037E-03	-1.15045E-03	1.75126E-03	-138.53
3

THIRD ORDER: FREQUENCY(1, 2, -1) = 3.250E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	4.59982E-05	-1.78787E-04	1.84861E-04	-7.46631E+01	7.04972E-05	-2.69181E-04	2.72293E-04	-75.27
2	6.35931E-04	-7.37985E-04	9.74215E-04	-6.02269E+01	9.53972E-04	-1.10698E-03	1.45133E-03	-49.25
3

THIRD ORDER: FREQUENCY(1, 2, -2) = 3.000E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	4.32729E-03	-1.46950E-04	1.54675E-04	-7.62116E+01	7.24094E-05	-2.20424E-04	2.22013E-04	-71.81
2	6.56543E-04	-6.79441E-04	9.44822E-04	-6.04930E+01	9.84815E-04	-1.01916E-03	1.41723E-03	-45.58
3

THIRD ORDER: FREQUENCY(2, 2, 2) = 9.750E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.34157E-03	7.65901E-04	1.54480E-03	-5.62225E+01	-3.35392E-04	1.91475E-04	3.85200E-04	150.28
2	-1.88210E-03	-1.47422E-03	2.39074E-03	-5.24294E+01	-4.70526E-04	-3.68555E-04	5.97635E-04	-141.93
3

THIRD ORDER: FREQUENCY(2, 2, -1) = 3.500E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	4.11550E-05	-2.10797E-04	2.14777E-04	-7.33602E+01	3.08652E-05	-1.58098E-04	1.61083E-04	-78.95
2	6.16960E-04	-7.99287E-04	1.00970E-03	-5.99161E+01	4.62720E-04	-5.99485E-04	7.57277E-04	-52.34
3

THIRD ORDER: FREQUENCY(2, 2, -2) = 3.250E+06 HZ

TRANSFER FUNCTION				OUTPUT VOLTAGE				
PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	4.11550E-05	-2.10797E-04	2.14777E-04	-7.33602E+01	3.08652E-05	-1.58098E-04	1.61083E-04	-78.95
2	6.16960E-04	-7.99287E-04	1.00970E-03	-5.99161E+01	4.62720E-04	-5.99485E-04	7.57277E-04	-52.34
3

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	4.55150E-05	-1.70850E-04	1.76809E-04	-7.50499E+01	3.41363E-05	-1.28138E-04	1.32607E-04	-75.08
3	6.24579E-04	-7.20036E-04	9.53179E-04	-6.04165E+01	4.68434E-04	-5.40027E-04	7.14885E-04	-49.05

INPUT FREQUENCIES:
FREQUENCY VALUE(HZ)

1 2.300E+07
2 2.325E+07

FIRST ORDER:

FREQUENCY(1) = 2.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	3.71975E-01	1.20031E-01	3.90830E-01	-8.15913E+00	3.71975E-01	1.20031E-01	3.90830E-01	17.89
3	-9.00586E-01	9.28940E-01	1.29331E+00	2.23743E+00	-9.00586E-01	9.28940E-01	1.29331E+00	134.11

FIRST ORDER:

FREQUENCY(2) = 2.325E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	4.00936E-01	1.04933E-01	4.14155E-01	-7.65652E+00	4.00936E-01	1.04933E-01	4.14155E-01	14.59
3	-9.22164E-01	1.04472E+00	1.39343E+00	2.88209E+00	-9.22164E-01	1.04472E+00	1.39343E+00	131.43

SECOND ORDER:

FREQUENCY(1, 1) = 4.600E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	-1.81737E-03	-2.59353E-02	2.59157E-02	-3.16553E+01	-5.08333E-04	-1.23979E-02	1.30078E-02	-92.34
3	6.29263E-02	-5.81745E-02	8.56391E-02	-2.13405E+01	3.14645E-02	-2.90873E-02	4.28456E-02	-43.75

SECOND ORDER:

FREQUENCY(1, 2) = 4.625E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	-2.20123E-03	-2.77653E-02	2.78534E-02	-3.11024E+01	-2.20123E-03	-2.77653E-02	2.78534E-02	-91.53
3	5.31537E-02	-6.35062E-02	8.5483E-02	-2.09588E+01	6.31337E-02	-6.35062E-02	8.55463E-02	-43.17

SECOND ORDER:

FREQUENCY(1, -1) = 0 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	-1.11713E-14	0	1.11713E-14	-2.79038E+02	-1.11713E-14	0	1.11713E-14	180.00

3 7.62512E-02 0 7.62512E-02 -2.23551E+01 0 7.62512E-02 0

SECOND ORDER:

FREQUENCY(2, 2) = 4.650E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	-3.55585E-03	-2.96051E-02	2.98179E-02	-3.05105E+01	-1.77793E-03	-1.48026E-02	1.49090E-02	-96.85
3	6.31191E-02	-6.91066E-02	9.35935E-02	-2.05751E+01	3.15595E-02	-3.45533E-02	4.67988E-02	-47.59

SECOND ORDER:

FREQUENCY(2, -1) = 2.500E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	1.32285E-05	-2.97124E-05	3.25242E-05	-8.97559E+01	1.32285E-05	-2.97124E-05	3.25242E-05	-65.00
3	8.12035E-02	-3.48884E-03	8.12804E-02	-2.18003E+01	8.12035E-02	-3.48884E-03	8.12804E-02	-2.46

SECOND ORDER:

FREQUENCY(2, -2) = 0 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	-1.05899E-14	0	1.05899E-14	-2.79502E+02	-1.05899E-14	0	1.05899E-14	180.00
3	8.66037E-02	0	8.66037E-02	-2.12493E+01	8.66037E-02	0	8.66037E-02	0

THIRD ORDER:

FREQUENCY(1, 1, 1) = 6.900E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	-3.44025E-02	3.05333E-02	4.69157E-02	-2.67419E+01	-8.60066E-03	7.63995E-03	1.15035E-02	138.39
3	-3.72443E-02	-1.30146E-02	3.94327E-02	-2.80785E+01	-9.31107E-03	-3.25365E-03	9.65318E-03	-150.74

THIRD ORDER:

FREQUENCY(1, 1, 2) = 6.925E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
2	-3.45333E-02	3.44702E-02	4.91220E-02	-2.61745E+01	-2.62476E-02	2.58526E-02	3.68415E-02	135.43
3	-4.01310E-02	-1.18345E-02	4.18398E-02	-2.75682E+01	-3.00982E-02	-6.87598E-03	3.13757E-02	-163.57

THIRD ORDER:

FREQUENCY(1, 1, -1) = 2.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-2.57602E-02	1.77800E-01	1.79636E-01	-1.49112E+01	-1.93202E-02	1.33350E-01	1.34742E-01	93.24
2	-4.24884E-01	-3.95346E-01	5.80332E-01	-4.72617E+00	-3.18648E-01	-2.96510E-01	4.35264E-01	-137.06
3								

THIRD ORDER:

FREQUENCY(1, 1, -2) = 2.275E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-4.15943E-02	1.77226E-01	1.82039E-01	-1.47967E+01	-3.11883E-02	1.32919E-01	1.36529E-01	103.21
2	-3.91277E-01	-4.32673E-01	5.83355E-01	-4.68134E+00	-2.93458E-01	-3.24504E-01	4.37516E-01	-133.12
3								

THIRD ORDER:

FREQUENCY(1, 2, 2) = 6.950E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-3.54152E-02	3.86695E-02	5.24344E-02	-2.56077E+01	-2.55614E-02	2.90002E-02	3.93253E-02	133.43
2	-4.31277E-02	-1.04345E-02	4.43720E-02	-2.70978E+01	-3.23458E-02	-7.82587E-03	3.32780E-02	-163.40
3								

THIRD ORDER:

FREQUENCY(1, 2, -1) = 2.325E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.04931E-02	2.01461E-01	2.01734E-01	-1.39044E+01	-1.57471E-02	3.02192E-01	3.02601E-01	93.53
2	-5.20045E-01	-4.01564E-01	6.57033E-01	-3.64818E+00	-7.60067E-01	-6.02346E-01	9.63553E-01	-143.33
3								

THIRD ORDER:

FREQUENCY(1, 2, -2) = 2.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-2.52050E-02	2.01948E-01	2.04039E-01	-1.38053E+01	-4.38090E-02	3.02922E-01	3.06074E-01	93.23
2	-4.82667E-01	-4.48907E-01	6.59154E-01	-3.62026E+00	-7.24000E-01	-6.73360E-01	9.69731E-01	-137.06
3								

THIRD ORDER:

FREQUENCY(2, 2, 2) = 5.975E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2								
3								

2 -3.56312E-02 4.31591E-02 5.59669E-02 -2.50414E+01 -8.90781E-03 1.07899E-02 1.39917E-02 129.54
 3 -4.62296E-02 -8.79537E-03 4.70589E-02 -2.65472E+01 -1.15574E-02 -2.19884E-03 1.17647E-02 -169.23

THIRD ORDER:

FREQUENCY(2, 2,-1)= 2.350E+07 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.03286E-02	2.26800E-01	2.26835E-01	-1.28658E+01	7.74649E-03	1.69950E-01	1.70127E-01	87.39
2	-6.31461E-01	-3.95342E-01	7.45009E-01	-2.55677E+00	-4.73596E-01	-2.96506E-01	5.58757E-01	-147.93
3

THIRD ORDER:

FREQUENCY(2, 2,-2)= 2.325E+07 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.13757E-02	2.28801E-01	2.29109E-01	-1.27992E+01	-8.90679E-03	1.71600E-01	1.71821E-01	89.97
2	-5.90709E-01	-4.55944E-01	7.45205E-01	-2.54283E+00	-4.43032E-01	-3.41938E-01	5.59654E-01	-142.34
3

INPUT FREQUENCIES:
FREQUENCY VALUE (HZ)

1 4.300E+07
2 4.353E+07

FIRST ORDER:

FREQUENCY(1) = 4.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-4.03063E-02	-3.03914E-02	5.70741E-02	-2.43712E+01	-4.83063E-02	-3.03914E-02	5.70741E-02	-147.93
2	5.17760E-01	-3.97536E-02	5.19284E-01	-5.59190E+00	5.17760E-01	-3.97536E-02	5.19284E-01	-4.53

FIRST ORDER:

FREQUENCY(2) = 4.353E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-4.75550E-02	-2.06801E-02	5.1033E-02	-2.50193E+01	-4.75550E-02	-2.06801E-02	5.1033E-02	-147.13
2	5.14007E-01	-4.04186E-02	5.15674E-01	-5.75230E+00	5.14007E-01	-4.04186E-02	5.15674E-01	-4.53

SECOND ORDER:

FREQUENCY(1, 1) = 8.500E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.43140E-02	-4.1037E-04	1.53043E-03	-5.62535E+01	7.40700E-04	-2.00444E-04	7.59743E-04	-13.20
2	1.43140E-02	1.72035E-04	1.43013E-03	-5.65177E+01	7.41574E-04	8.64175E-05	7.40533E-04	9.53

SECOND ORDER:

FREQUENCY(1, 2) = 8.525E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.43101E-02	-1.1013E-04	1.51925E-03	-5.53334E+01	1.43101E-03	-4.18139E-04	1.51925E-03	-13.01
2	1.43025E-02	1.53337E-04	1.43143E-03	-5.65276E+01	1.43025E-03	1.56337E-04	1.49145E-03	9.02

SECOND ORDER:

FREQUENCY(1, 1) = 0 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-2.53705E-15	0	2.53705E-15	-2.91479E+02	-2.53705E-15	0	2.53705E-15	180.00

3 3.54558E-03 0 3.54558E-03 -4.90063E+01 3.54558E-03 0 3.54558E-03 0

SECOND ORDER:

FREQUENCY(2, 2) = 8.650E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.44037E-03	-4.1932E-04	1.50058E-03	-5.64744E+01	7.20433E-04	-2.09696E-04	7.5030E-04	-19.23
2	1.48357E-03	1.40009E-04	1.49016E-03	-5.65354E+01	7.41783E-04	7.00045E-05	7.45075E-04	5.39

SECOND ORDER:

FREQUENCY(2, -1) = 2.500E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	6.57249E-07	-1.35254E-06	1.50378E-06	-1.16456E+02	6.57249E-07	-1.35254E-06	1.50378E-06	-84.03
3	3.50232E-03	1.11219E-05	3.50234E-03	-4.91128E+01	3.50232E-03	1.11219E-05	3.50234E-03	.10

SECOND ORDER:

FREQUENCY(2, -2) = 0 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-2.62314E-15	0	2.62314E-15	-2.91624E+02	-2.62314E-15	0	2.62314E-15	100.00
3	3.45599E-03	0	3.45599E-03	-4.92285E+01	3.45599E-03	0	3.45599E-03	0

THIRD ORDER:

FREQUENCY(1, 1, 1) = 1.290E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	3.78349E-04	3.47002E-04	5.13379E-04	-6.57912E+01	9.45871E-05	8.67505E-05	1.23345E-04	43.03
3	-8.65553E-05	1.62492E-04	1.83253E-04	-7.46447E+01	-2.22415E-05	4.06230E-05	4.63122E-05	113.79

THIRD ORDER:

FREQUENCY(1, 1, 2) = 1.293E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	3.74955E-04	3.41040E-04	5.09854E-04	-6.59023E+01	2.81217E-04	2.55780E-04	3.80140E-04	42.39
3	-3.71743E-05	1.60486E-04	1.62616E-04	-7.47692E+01	-6.53507E-05	1.20349E-04	1.36855E-04	113.81

THIRD ORDER:

FREQUENCY(1, 1, -1) = 4.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.41012E-04	-3.57405E-04	3.93537E-04	-6.81003E+01	-1.05759E-04	-2.75554E-04	2.95153E-04	-111.69
2	7.15363E-04	-7.57775E-04	1.04210E-03	-5.96418E+01	5.35522E-04	-5.69331E-04	7.81575E-04	-43.55
3								

THIRD ORDER:

FREQUENCY(1, 1, -2) = 4.275E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.40734E-04	-3.51729E-04	3.83141E-04	-6.82202E+01	-1.05550E-04	-2.71287E-04	2.91105E-04	-111.23
2	7.19361E-04	-7.57143E-04	1.04439E-03	-5.96228E+01	5.35520E-04	-5.67897E-04	7.83291E-04	-43.47
3								

THIRD ORDER:

FREQUENCY(1, 2, 2) = 1.295E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	3.71582E-04	3.35168E-04	5.00411E-04	-6.50135E+01	2.78623E-04	2.51376E-04	3.75295E-04	41.65
2	-8.54171E-05	1.58456E-04	1.80021E-04	-7.48936E+01	-6.40620E-05	1.18849E-04	1.55016E-04	110.50
3								

THIRD ORDER:

FREQUENCY(1, 2, -1) = 4.325E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.37836E-04	-3.53321E-04	3.59143E-04	-6.81977E+01	-2.06754E-04	-5.45881E-04	5.83726E-04	-110.74
2	6.94712E-04	-7.39409E-04	1.01457E-03	-5.98744E+01	1.04207E-03	-1.10911E-03	1.52165E-03	-43.47
3								

THIRD ORDER:

FREQUENCY(1, 2, -2) = 4.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.37368E-04	-3.58153E-04	3.83593E-04	-6.83226E+01	-2.06053E-04	-5.37229E-04	5.75265E-04	-110.50
2	6.57472E-04	-7.38434E-04	1.01575E-03	-5.98642E+01	1.04921E-03	-1.10765E-03	1.52365E-03	-43.53
3								

THIRD ORDER:

FREQUENCY(2, 2, 2) = 1.298E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2								
3								

2 3.62325E-04 3.2333E-04 4.94049E-04 -6.91246E+01 -6.91246E-04 8.23462E-05 1.23512E-04 41.81
 3 -3.6633E-05 1.56432E-04 1.77463E-04 -7.50177E+01 -7.50177E-04 3.51229E-05 4.43665E-05 116.14

THIRD ORDER: FREQUENCY(2, 2, -1) = 4.350E+07 HZ
 TRANSFER FUNCTION
 OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.2433E-04	-3.50457E-04	3.3433E-04	-6.8340E+01	-1.01126E-04	-2.70350E-04	2.8254E-04	-110.81
3	6.74310E-04	-7.2161E-04	9.37975E-04	-8.01051E+01	5.06103E-04	-5.41210E-04	7.40551E-04	-63.92

THIRD ORDER: FREQUENCY(2, 2, -2) = 4.325E+07 HZ
 TRANSFER FUNCTION
 OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.2110E-04	-3.5431E-04	3.7913E-04	-6.84237E+01	-1.00840E-04	-2.65550E-04	2.8429E-04	-110.70
3	6.7637E-04	-7.2001E-04	9.3303E-04	-8.01941E+01	5.07263E-04	-5.40226E-04	7.41065E-04	-63.80

INPUT FREQUENCIES:
FREQUENCY VALUE (HZ)
1 6.300E+07
2 6.325E+07

FIRST ORDER:

FREQUENCY(1) = 6.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	-2.05059E-02	-5.37372E-03	2.11983E-02	-3.34740E+01	-2.05059E-02	-5.37372E-03	2.11983E-02	-163.62
3	3.82162E-01	-6.5521E-02	3.87711E-01	-8.22984E+00	3.82162E-01	-6.5521E-02	3.87711E-01	-9.11

FIRST ORDER:

FREQUENCY(2) = 6.325E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	-2.02522E-02	-5.30130E-03	2.10375E-02	-3.35395E+01	-2.02522E-02	-5.30130E-03	2.10375E-02	-163.60
3	3.81355E-01	-6.55742E-02	3.86882E-01	-8.24595E+00	3.81355E-01	-6.55742E-02	3.86882E-01	-9.11

SECOND ORDER:

FREQUENCY(1, 1) = 1.260E+03 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	2.85703E-04	-2.65579E-04	3.90075E-04	-6.81770E+01	1.42851E-04	-1.32785E-04	1.55037E-04	-43.91
3	1.41430E-03	-1.01825E-03	1.74313E-03	-5.51734E+01	7.07402E-04	-5.09125E-04	8.71555E-04	-45.14

SECOND ORDER:

FREQUENCY(1, 2) = 1.263E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	2.83331E-04	-2.54674E-04	3.87722E-04	-6.82295E+01	2.83331E-04	-2.64574E-04	3.87722E-04	-43.93
3	1.41321E-03	-1.02200E-03	1.74403E-03	-5.51689E+01	1.41321E-03	-1.02200E-03	1.74403E-03	-45.07

SECOND ORDER:

FREQUENCY(1, -1) = 0 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2	-8.31112E-16	0	8.31112E-16	-3.01607E+02	-8.31112E-16	0	8.31112E-16	169.00

3 9.13124E-04 0 9.13124E-04 -6.07894E+01 9.13124E-04 0 9.13124E-04 0

SECOND ORDER:

FREQUENCY(2, 2) = 1.265E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	2.80975E-04	-2.63768E-04	3.82821E+01	-6.82821E+01	1.40487E-04	-1.31884E-04	1.52692E-04	-43.19
2	1.41161E-03	-1.02574E-03	1.74493E-03	-5.51644E+01	7.05805E-04	-5.12871E-04	8.72466E-04	-35.00

SECOND ORDER:

FREQUENCY(2, -1) = 2.500E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.68576E-08	-3.33644E-07	3.34069E-07	-1.29523E+02	1.68576E-08	-3.33644E-07	3.34069E-07	-87.11
3	9.08266E-04	5.12523E-06	9.08280E-04	-6.08356E+01	9.08266E-04	5.12523E-06	9.08280E-04	.32

SECOND ORDER:

FREQUENCY(2, -2) = 0 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-8.20584E-16	0	8.20584E-16	-3.01718E+02	-8.20584E-16	0	8.20584E-16	180.00
2	9.02404E-04	0	9.02404E-04	-6.08920E+01	9.02404E-04	0	9.02404E-04	0

THIRD ORDER:

FREQUENCY(1, 1, 1) = 1.890E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	6.48898E-05	-2.63355E-05	6.50430E-05	-8.37350E+01	1.62474E-05	-6.58413E-07	1.62607E-05	-2.32
3	2.98444E-05	2.66936E-05	2.68599E-05	-9.14179E+01	7.46111E-07	6.67339E-06	6.71497E-06	83.62

THIRD ORDER:

FREQUENCY(1, 1, 2) = 1.893E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	5.48853E-05	-2.79758E-06	6.46452E-05	-8.37892E+01	4.84390E-05	-2.09818E-06	4.84844E-05	-2.48
3	3.02881E-05	2.65831E-05	2.67556E-05	-9.14517E+01	2.27461E-06	1.99373E-05	2.00667E-05	83.49

THIRD ORDER:

FREQUENCY(1, 1, -1) = 6.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.94210E-05	-5.54933E-05	5.37973E-05	-8.46128E+01	-1.45732E-05	-4.16202E-05	4.48001E-05	-100.00
3	2.05506E-05	-4.16511E-05	4.64451E-05	-8.66612E+01	1.54130E-05	-3.12333E-05	3.48360E-05	-83.74

THIRD ORDER:

FREQUENCY(1, 1, 2) = 5.275E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.92502E-05	-5.50321E-05	5.33013E-05	-8.45394E+01	-1.44272E-05	-4.12741E-05	4.27633E-05	-100.00
3	2.07022E-05	-4.18006E-05	4.56467E-05	-8.66336E+01	1.55270E-05	-3.13507E-05	3.48560E-05	-83.74

THIRD ORDER:

FREQUENCY(1, 2, 2) = 1.335E+03 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	5.41835E-05	-2.55332E-05	6.42511E-05	-9.38424E+01	4.81872E-05	-2.21530E-05	4.01501E-05	-73.54
3	2.03072E-05	2.64734E-05	2.86521E-05	-9.14842E+01	2.31055E-05	1.55551E-05	1.55551E-05	90.15

THIRD ORDER:

FREQUENCY(1, 2, 1) = 5.325E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.54063E-05	-5.52419E-05	5.36433E-05	-8.46353E+01	-2.81053E-05	-6.30128E-05	8.75663E-05	-100.00
3	2.01572E-05	-4.10458E-05	4.57283E-05	-8.67933E+01	3.02358E-05	-6.15538E-05	6.85824E-05	-83.04

THIRD ORDER:

FREQUENCY(1, 2, 2) = 5.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.81593E-05	-5.42452E-05	5.31093E-05	-8.47151E+01	-2.87593E-05	-6.22689E-05	8.71640E-05	-100.00
3	2.01033E-05	-4.10952E-05	4.57513E-05	-8.67919E+01	3.01524E-05	-6.16427E-05	6.85824E-05	-83.00

THIRD ORDER:

FREQUENCY(2, 2, 2) = 1.837E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...

2 6.37225E-05 -3.11893E-05 6.38587E-05 -8.38956E+01 -7.79733E-07 1.59456E-05 1.59647E-05 -2.80
 3 3.12836E-06 2.63643E-05 2.65494E-05 -9.15189E+01 6.59112E-06 6.63736E-06 6.63736E-06 83.23

THIRD ORDER:

FREQUENCY(2, 2, -1) = 6.350E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.93554E-05	-5.51990E-05	5.85077E-05	-8.48557E+01	-1.45473E-05	-4.13992E-05	4.38808E-05	-109.35
3	1.97707E-05	-4.04439E-05	4.50222E-05	-8.69315E+01	1.48280E-05	-3.03357E-05	3.37666E-05	-63.95

THIRD ORDER:

FREQUENCY(2, 2, -2) = 6.325E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.91509E-05	-5.43707E-05	5.79312E-05	-3.47117E+01	-1.43707E-05	-4.10020E-05	4.34334E-05	-103.31
3	1.65340E-05	-4.05531E-05	4.43686E-05	-3.69311E+01	1.46433E-05	-3.02886E-05	3.36516E-05	-54.21

INPUT FREQUENCIES:
FREQUENCY VALUE(HZ)
1 8.300E+07
2 8.325E+07

FIRST ORDER:

FREQUENCY(1) = 8.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	-1.16963E-02	-4.50892E-05	1.16364E-02	-3.86390E+01	-1.16963E-02	-4.50892E-05	1.16364E-02	-179.78
2	3.41029E-01	-8.15271E-02	3.50638E-01	-9.10281E+00	3.41029E-01	-8.15271E-02	3.50638E-01	-13.44

FIRST ORDER:

FREQUENCY(2) = 8.325E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	-1.16173E-02	-1.54330E-05	1.16174E-02	-3.86397E+01	-1.16173E-02	-1.54330E-05	1.16174E-02	-179.82
2	3.40672E-01	-8.17218E-02	3.50336E-01	-9.11030E+00	3.40672E-01	-8.17218E-02	3.50336E-01	-13.49

SECOND ORDER:

FREQUENCY(1, 1) = 1.660E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	7.65070E-05	-1.50451E-04	1.73333E-04	-7.50455E+01	7.65070E-05	-1.50451E-04	1.73333E-04	-84.34
2	1.06320E-03	-1.41614E-03	1.77486E-03	-5.50197E+01	1.06320E-03	-1.41614E-03	1.77486E-03	-52.95

SECOND ORDER:

FREQUENCY(1, 2) = 1.662E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	7.52330E-05	-1.50230E-04	1.76157E-04	-7.50315E+01	7.52330E-05	-1.50230E-04	1.76157E-04	-84.47
2	1.06640E-03	-1.41762E-03	1.77555E-03	-5.50212E+01	1.06640E-03	-1.41762E-03	1.77555E-03	-53.05

SECOND ORDER:

FREQUENCY(1, 1) = 0 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
1	-3.34320E-15	0	3.34320E-15	-3.09517E+02	-3.34320E-15	0	3.34320E-15	180.00
2	-3.34320E-15	0	3.34320E-15	-3.09517E+02	-3.34320E-15	0	3.34320E-15	180.00

3 4.25217E-04 0 4.25217E-04 -6.74278E+01 4.25217E-04 0 4.25217E-04 0

SECOND ORDER:

FREQUENCY(2, 2) = 1.565E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	7.52723E-05	-1.58470E-04	1.75439E-04	-7.51175E+01	3.76364E-05	-7.92349E-05	8.77195E-05	-94.59
2	1.06390E-03	-1.41912E-03	1.77384E-03	-5.50227E+01	5.31950E-04	-7.09390E-04	8.65816E-04	-53.14

SECOND ORDER:

FREQUENCY(2, -1) = 2.500E+05 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.08379E-07	-1.35037E-07	1.73931E-07	-1.35192E+02	-1.06379E-07	-1.26037E-07	1.73931E-07	-120.54
2	4.23733E-04	3.30224E-05	4.23745E-04	-6.74579E+01	4.23733E-04	3.30224E-05	4.23745E-04	-45

SECOND ORDER:

FREQUENCY(2, -2) = 0 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-3.30951E-16	0	3.30951E-16	-3.08605E+02	-3.30951E-16	0	3.30951E-16	180.00
2	4.21860E-04	0	4.21860E-04	-6.74966E+01	4.21860E-04	0	4.21860E-04	0

THIRD ORDER:

FREQUENCY(1, 1, 1) = 2.490E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.59721E-05	-1.21057E-05	1.93318E-05	-3.42746E+01	3.76803E-06	-3.02843E-06	4.85283E-06	-38.77
2	9.09123E-05	1.24541E-05	1.54274E-05	-9.62341E+01	2.27282E-06	3.11602E-06	3.85563E-06	53.63

THIRD ORDER:

FREQUENCY(1, 1, 2) = 2.493E+08 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	1.49771E-05	-1.20919E-05	1.92491E-05	-3.43118E+01	1.12388E-05	-9.06890E-06	1.44353E-05	-33.52
2	9.10429E-05	1.24283E-05	1.54062E-05	-9.62461E+01	6.82282E-06	9.32115E-06	1.15543E-05	53.73

THIRD ORDER:

FREQUENCY(1, 1, -1) = 8.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.07529E-05	-1.73825E-05	2.04335E-05	-9.37906E+01	-8.06497E-06	-1.50369E-05	1.55207E-05	-121.74
2	-1.10115E-05	2.24383E-05	1.12372E-05	-9.89864E+01	-8.25862E-06	1.68287E-05	8.45234E-05	133.43
3								

THIRD ORDER:

FREQUENCY(1, 1, -2) = 8.275E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.46341E-05	-1.73325E-05	2.03510E-05	-9.38406E+01	-8.01311E-06	-1.25594E-05	1.52707E-05	-121.23
2	-1.10022E-05	2.12464E-05	1.12054E-05	-9.90114E+01	-8.25163E-06	1.59348E-05	8.40408E-05	133.07
3								

THIRD ORDER:

FREQUENCY(1, 2, 2) = 2.495E+03 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.46325E-05	-1.20778E-05	1.91697E-05	-9.43490E+01	1.11618E-05	-9.05337E-06	1.43751E-05	-33.05
2	9.11724E-06	1.23925E-05	1.53850E-05	-9.62580E+01	6.83793E-06	9.25440E-06	1.15200E-05	55.03
3								

THIRD ORDER:

FREQUENCY(1, 2, -1) = 8.325E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.07545E-05	-1.73030E-05	2.03722E-05	-9.38189E+01	-1.61318E-05	-2.58545E-05	3.05524E-05	-121.23
2	-1.09231E-05	2.35197E-05	1.12321E-05	-9.89907E+01	-1.64747E-05	3.52735E-05	1.62463E-05	137.01
3								

THIRD ORDER:

FREQUENCY(1, 2, -2) = 8.300E+07 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...	-1.06782E-05	-1.72435E-05	2.02824E-05	-9.38576E+01	-1.60182E-05	-2.58532E-05	3.04235E-05	-121.77
2	-1.10580E-05	2.28945E-05	1.12323E-05	-9.89442E+01	-1.65870E-05	3.43418E-05	1.69388E-05	133.30
3								

THIRD ORDER:

FREQUENCY(2, 2, 2) = 2.498E+03 HZ

TRANSFER FUNCTION

OUTPUT VOLTAGE

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...								
2								
3								

2 1.47884E-05
3 9.13013E-05

1.20335E-05
1.23585E-05

1.90847E-05
1.53640E-05

-9.43863E+01
-9.62899E+01

3.69711E-06
2.28253E-06

-3.01590E-05
3.08922E-05

4.77115E-03
3.84100E-03

-33.21
53.54

THIRD ORDER:

FREQUENCY(2, 2, -1) = 8.350E+07 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.07551E-05	-1.72237E-05	2.03054E-05	-9.33474E+01	-8.06705E-06	-1.29178E-05	1.62102E-05	-121.83
3	-1.09345E-05	2.45907E-05	1.12271E-05	-9.89947E+01	-8.21585E-06	1.84430E-05	8.42601E-05	157.35

THIRD ORDER:

FREQUENCY(2, 2, -2) = 8.325E+07 HZ

TRANSFER FUNCTION

PORT NO	REAL	IMAGINARY	MAGNITUDE	20LOG MAG	REAL	IMAGINARY	MAGNITUDE	PHASE DEG
...
2	-1.06732E-05	-1.71545E-05	2.02035E-05	-9.38313E+01	-8.00493E-06	-1.28350E-05	1.51111E-05	-121.03
3	-1.11120E-05	2.43311E-05	1.15305E-05	-9.88768E+01	-8.25472E-06	1.82330E-05	8.50036E-05	157.55

TIME FOR FORMING ZOO(SEC) 3.9170
TIME FOR OBTAINING OUTPUT SPECTRUM(SEC) 13.0210
TOTAL EXECUTION TIME(SEC) 21.9530

*** P R A N C ***
SEPTEMBER 1979 VERSION

CHAPTER 5

PROGRAMMER'S GUIDE FOR PRANC

5-1. Introduction

The ideas presented in Chapters 2 and 3 have been used to adapt the Volterra series method for computer-aided distortion analysis of circuits with polynomial type nonlinear elements. PRANC (Program for Analyzing Nonlinear Circuits), a digital computer program written in FORTRAN IV, is the outcome of this effort. This chapter presents in detail the program structure and the description of the subprograms contained in PRANC. This chapter should be most useful for programmers wishing to modify the program.

Section 5-2 presents the program structure of PRANC. By pointing out the sequence of phases that are involved in a typical analysis run, the interaction between the various subprograms is depicted. The discussion in this section provides an insight into how the "equivalencing" of arrays should be carried out for conserving storage.

Section 5-3 presents the details of each subprogram contained in PRANC. These details include: 1) brief description; 2) glossary of FORTRAN variables; and 3) listing of each subprogram. The contents of this section should aid the programmer in making future modifications to the program.

PRANC has been developed on the CDC 6500/6600 computer at the Purdue University computing facility, and as such certain machine- and library-dependent instructions exist. These system dependent cards in the program are listed in section 5-4. Cards capable of calling equivalent functions can be substituted in their place for adapting the program on a different system.

5-2. Program Structure of PRANC

Before detailing the program structure of PRANC, it is instructive to delineate the sequence of steps that are involved in a typical analysis run when using our computational algorithms. The program structure and its modularity are better understood once a knowledge of the sequence of steps has been acquired. Referring to a collection of steps as a phase, the following is a partitioning of phases that are involved in a typical analysis run:

Phase A: The following functions are performed during this phase:

- (a) Read input data
- (b) Control the interaction between the other phases.

In a sense, this phase can be regarded to extend during the entire analysis run.

Phase B: The user desired options are scanned during this phase and the flag variable associated with each option is appropriately set.

Phase C: This phase is responsible for the following functions:

- (a) Setting up of the arrays for the network description in a prescribed manner.
- (b) Assigning addresses based on the user-specified options and the nonlinear element topology.

Phase D: The Hybrid analysis, which yields the constraint matrix [20], is performed during this phase.

Phase E: The state space representation for the linearized circuit is obtained during this phase.

Phase F: The eigenvalue-eigenvector information is determined from the state space description during this phase.

Phase G: The printing and the formation of the entries of the open-circuit impedance matrix is carried out during this phase.

Phase H: The first-, second-, and third-order transfer functions are computed during this phase.

Phase I: The following functions are performed during this phase:

- (a) Compute the output voltages at each frequency point from the transfer function values.
- (b) Print both the transfer function and the output voltage values at each discrete frequency point for the requested outputs.

Phase J: During this phase, the complete output spectrum at the user-requested port is printed and plotted.

Phase K: When a frequency sweep capability is requested, this phase is used to perform the said operation.

Phase L: When devices, such as transistors, diodes, etc., are to be represented by equivalent nonlinear models, this phase is used to calculate the parameters of the nonlinearities.

Several subroutines are required to perform the functions belonging to each of the aforementioned phases. PRANC, in its present version, consists of thirty-six sub-programs, whose interaction is depicted in Fig. 5-1. In order to provide a link between a phase and its associated sub-programs, the naming of the subroutines has been done in a deterministic manner: the first letter of the subroutine name signifies the phase to which it belongs.

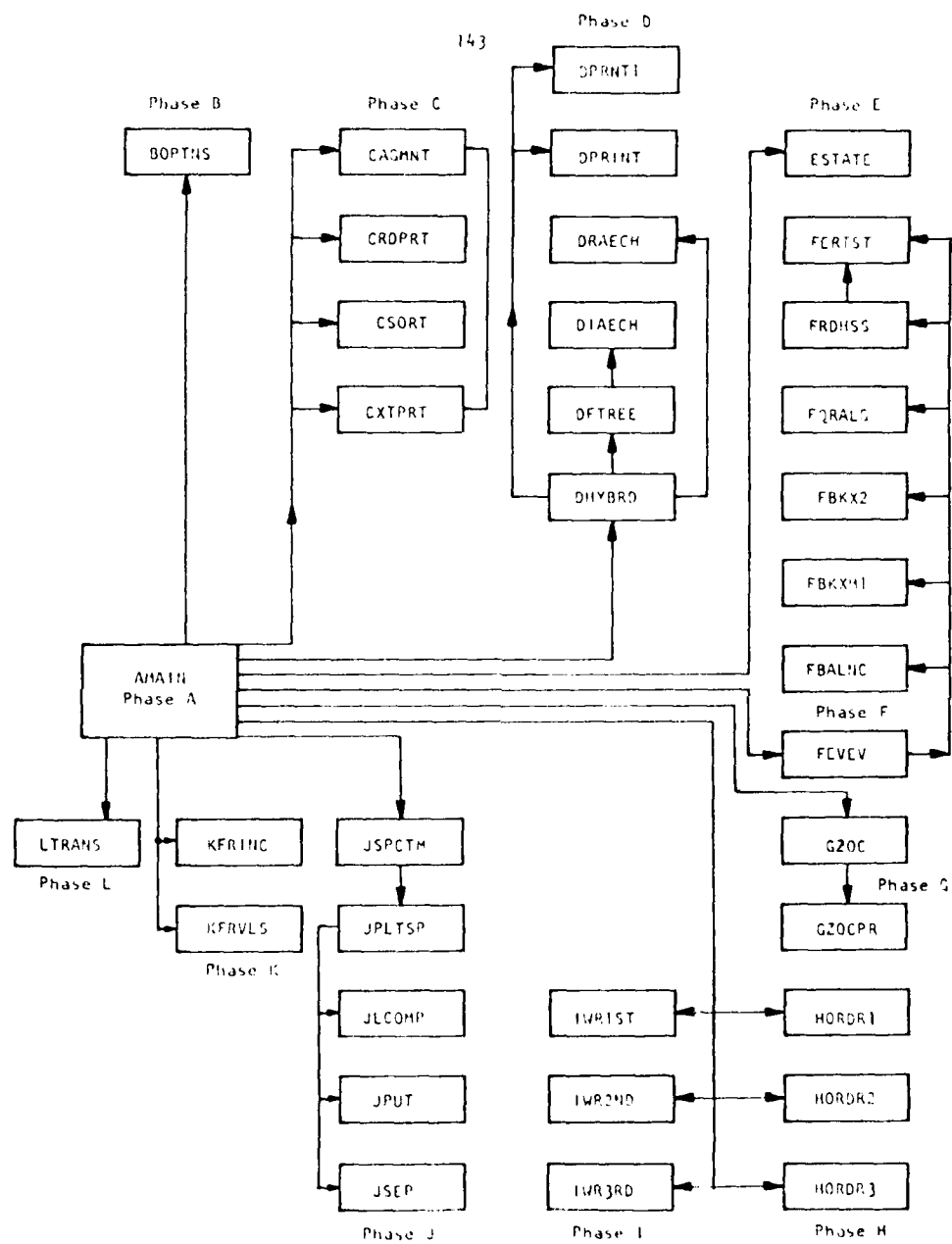


Fig. S-1. Program Structure of PRANC

Thus, for example, the subroutines HORDR1, HORDR2, HORDR3 perform the functions outlined under phase H.

In the following paragraphs the function of each subprogram appearing in Fig. 5-1 is outlined:

Program AMAIN is the executive calling program of PRANC.

Subroutine BOPTNS deciphers the desired user options.

Subroutine CAGMNT forms the augmented linear network by lumping the linear parts of the nonlinear elements with the existing linear network.

Subroutine CRDPRT identifies and combines the parallel energy storage elements and current sources appearing in the augmented linear network, thus effectively reducing the number of ports extracted for hybrid analysis.

Subroutine CSORT sorts the elements of the augmented linear network and arranges them in an order suitable for choosing a proper tree [20].

Subroutine CXTprt adds a branch to the linear network.

Subroutine DFTREE finds the proper tree from the incidence matrix [20].

Subroutine DHYBRD is the executive calling program for performing hybrid analysis of the augmented linear circuit to obtain the constraint matrix.

Subroutine DIAECH is used to manipulate the incidence matrix into echelon form.

Subroutine DPRINT prints the entire constraint matrix whenever the debug option is requested by the user.

Subroutine DPRNT1 prints only part of the constraint matrix describing the port equations whenever the debug option is requested by the user.

Subroutine DRAECH operates on the rows of the hybrid matrix to reduce it into echelon form.

Subroutine ESTATE formulates the state space description of the augmented linear network and, if desired by the user, prints this description.

Subroutine FBALNC balances the matrix whose eigenvalues are to be determined.

Subroutine FEVEV is the executive calling program used to determine the eigenvalues and their associated eigenvectors.

Subroutine FBKXM1 is used to back transform the eigenvectors of an Hessenberg matrix.

Subroutine FBKXM2 is used to back-transform the eigenvectors of a balanced matrix.

Subroutine FERTST is used to print the error diagnosis arising in eigenvalues-eigenvectors problems.

Subroutine FQRALG determines the eigenvalues and the eigenvectors of the Hessenberg matrix.

Subroutine FRDHSS reduces a matrix to the Hessenberg form.

Subroutine GZOC forms the matrices used to store the entries of the open circuit impedance matrix.

Subroutine GZOCPR prints the entries of the open circuit impedance matrix whenever desired by the user.

Subroutine HORDR1 computes the first-order transfer function at each positive and negative input frequency value.

Subroutine HORDR2 computes the second-order transfer function at each frequency combination appearing in the second-order output spectrum.

Subroutine HORDR3 computes the third-order transfer function at each non-negative frequency combination appearing in the third-order output spectrum.

Subroutine IWR1ST determines the first-order output spectrum and prints it along with the first-order transfer function at the user-specified output ports.

Subroutine IWR2ND determines the second-order output spectrum for non-negative frequencies and prints it along with the second-order transfer function values at the user-specified output ports.

Subroutine IWR3RD determines the third-order output spectrum for non-negative frequencies and prints it along with the third-order transfer function values at the user-specified output ports.

Subroutine JSPCTM performs histogram analysis of all output frequency components and combines the common-ones. It also prints and plots the complete output spectrum at the user-requested port, whenever desired.

Subroutine JPLTSP perform the actual plotting of the output spectrum.

Function JLCOMP locates the data points for plotting.

Function JPUT also locates the data points for plotting.

Subroutine JSEP separates the alphabets in the y-axis label for vertical printing.

Subroutine KFRNC computes the frequency increments for each input frequency whenever the frequency sweep capability is requested.

Subroutine KFRVLS computes the new frequency values during the frequency sweep.

Subroutine LTRANS computes the coefficients of the polynomials representing the nonlinear elements in a bipolar transistor.

5-3. Glossary and Subprogram Listings for PRANC

In this section we shall present the specific task of each sub-program, along with its listing. The glossary of important FORTRAN variable names is included in the sub-program listing.

5-3.1 Program AMAIN

Program AMAIN is the executive calling program of PRANC. Its primary function is to read and write input data, to form appropriate arrays for the augmented linear network description, and to assign appropriate addresses for subsequent use in forming the nonlinear current sources. The addressing array NCONT used in PRANC deserves some explanation.

Based on the network element types, and their associated KEY values, the elements of the augmented linear network are arranged in the following order:

1. Capacitors
2. VCVSs
3. CCVSs
4. Resistors
5. Inductors
6. VCCSs
7. CCCSs
8. Independent current sources

Such an arrangement is warranted for the selection of a proper tree and the formulation of hybrid and state equations. The number of independent current sources is equal to the number of extracted ports, p , for the augmented linear network.

Initially, as each input information card is read, a zero-valued current source is applied across each prescribed input source branch, output branch, nonlinear element branch, and nonlinear element characteristic controlling branch (in the dependent nonlinear element case). Each zero-valued current source signifies an extracted port. Associated with each extracted port is an index number, $NCONT$, starting from 1 to n ($n \geq p$). Clearly some of the initially extracted ports may be in parallel. The p -port augmented linear network is obtained after the parallel zero-valued current source branches in the n -port network are combined.

The extracted ports in the p -ports network has the following arrangement:

Input port	NCONT(1)
Output port 1	NCONT(2)
Output port 2	NCONT(3)
.	
.	
.	
Output port k	NCONT(k+1)
Nonlinear element #1 port	NCONT(k+2)
Nonlinear element #1 controlling port	
Nonlinear element #1 controlling port	
Nonlinear element #2 port	
Nonlinear element #2 controlling port	
Nonlinear element #2 controlling port	
.	
.	
.	
Nonlinear element #l port	
Nonlinear element #l controlling port	
Nonlinear element #l controlling port	NCONT(3l+k+1)

The array NCONT contains the port number for each of the above extracted ports. Thus, NCONT(1) contains the port number for the input source port; NCONT(2) for the first output port; NCONT(3) for the second output port; NCONT(k+1) for the k-th output port; and so on. It should be noted that the independent nonlinear elements are treated as special cases of dependent nonlinear elements. Thus, if NCONT(5) = 3 signifies

port number 3 for a nonlinear capacitor, then the locations NCONT(6) and NCONT(7) will also contain 3. In the case of a dependent nonlinear element, the number for the controlling ports will usually be different. It should be clear from the above discussion that, for a single input, k-output network with ℓ nonlinearities, the length of the array used is $(3\ell+k+1)$. The array NCONT plays an important role when the various order steady-state responses are computed.

5-3.2. PRANC Listing:

```

C*****AMN 10
C***** P R A N C *****AMN 20
C***** PROGRAM FOR ANALYZING NONLINEAR CIRCUITS *****AMN 30
C*****AMN 40
C***** SEPTEMBER 1979 VERSION *****AMN 50
C*****AMN 60
C*****AMN 70
C***** PROGRAM AMAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT) *****AMN 80
C*****AMN 90
C*****AMN 100
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTIONS: *****AMN 110
C***** 1. READ AND WRITE INPUT CIRCUIT DESCRIPTION. *****AMN 120
C***** 2. ACT AS THE EXECUTIVE CALLING PROGRAM FOR PRANC. *****AMN 130
C*****AMN 140
C***** THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINES: *****AMN 150
C***** 1. DOPRNS *****AMN 160
C***** 2. CASMIT,CKTPRT,CRDPRT,CSORT *****AMN 170
C***** 3. EHYDRD *****AMN 180
C***** 4. ESTATE *****AMN 190
C***** 5. FEVEU *****AMN 200
C***** 6. GEOD *****AMN 210
C***** 7. HORDR1, HORDR2, HORDR3 *****AMN 220
C***** 8. INRIST, INREND, INRORD *****AMN 230
C***** 9. JSEPDI *****AMN 240
C***** 10. KFRINC, KFRULS *****AMN 250
C***** 11. LTRANS *****AMN 260
C***** 12.*** SECOND *** (LIBRARY DEPENDENT ROUTINE) *****AMN 270
C*****AMN 280
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES: *****AMN 290
C***** BR(K) : BRANCH NUMBER K IN THE AUGMENTED LINEAR *****AMN 300
C***** NETWORK *****AMN 310
C***** NFRON(K) : <FROM> (+) NODE NUMBER FOR BRANCH NUMBER K *****AMN 320
C***** NTO(K) : <TO> (-) NODE NUMBER FOR BRANCH NUMBER K *****AMN 330
C***** TYPE(K) : BRANCH NUMBER K ELEMENT TYPE *****AMN 340
C***** ECONT(K) : CONTROLLING BRANCH NUMBER FOR BRANCH K *****AMN 350
C***** KEY(K) : KEY VALUE FOR BRANCH K *****AMN 360
C***** VAL(K) : ELEMENT VALUE FOR BRANCH K *****AMN 370
C***** NTYPE(K) : TYPE OF NONLINEAR ELEMENT K *****AMN 380
C***** COEFF(K,J) : POLYNOMIAL COEFFICIENT VALUE FOR NONLINEAR *****AMN 390
C***** ELEMENT K *****AMN 400
C***** FREQ(I) : I-TH INPUT FREQUENCY VALUE *****AMN 410
C***** AMP(I) : I-TH INPUT FREQUENCY AMPLITUDE *****AMN 420
C***** PHASE(I) : I-TH INPUT FREQUENCY PHASE *****AMN 430
C***** LUNIT : INPUT FREQUENCY UNIT (HZ OR RAD/SEC) *****AMN 440
C***** A : INCIDENCE MATRIX FOR THE AUGMENTED LINEAR *****AMN 450
C***** NETWORK *****AMN 460
C***** ANC : CONSTRAINT (HYBRID) MATRIX *****AMN 470
C***** HEADER : HENDING VECTOR FOR THE CONSTRAINT MATRIX *****AMN 480
C***** EXTM : WORK MATRIX USED IN HYBRID ANALYSIS *****AMN 490
C***** EXTR2 : WORK MATRIX USED IN HYBRID ANALYSIS *****AMN 500
C***** CMAT : MATRIX A OF THE STATE SPACE REPRESENTATION *****AMN 510
C***** DMAT : MATRIX B OF THE STATE SPACE REPRESENTATION *****AMN 520
C***** EMAT : MATRIX C OF THE STATE SPACE REPRESENTATION *****AMN 530
C***** FMAT : MATRIX D OF THE STATE SPACE REPRESENTATION *****AMN 540
C***** EIGVAL(I) : I-TH COMPLEX EIGENVALUE (NATURAL FREQUENCY) *****AMN 550
C***** EIGVTS : MODAL MATRIX FOR SIMILARITY TRANSFORMATION *****AMN 560
C***** EMAT : MATRIX OBTAINED FROM THE PRODUCT OF MODAL *****AMN 570
C***** MATRIX INVERSE AND DMAT *****AMN 580
C***** CMAT : MATRIX OBTAINED FROM THE PRODUCT OF CMAT AND *****AMN 590
C***** MODAL MATRIX *****AMN 600

```

```

C      *      WK1,WK2      : WORK ARRAYS                                *AMN  610
C      *      NSTPS(I)    : I-TH INPUT FREQUENCY NUMBER OF INCREMENTS *AMN  620
C      *      FRINC(I)    : I-TH INPUT FREQUENCY INCREMENT VALUE      *AMN  630
C      *      HFR(I)      : I-TH INPUT FREQUENCY HIGHEST VALUE        *AMN  640
C      *      Y1(P,I)     : PORT P FIRST-ORDER OUTPUT AT FREQ(I)      *AMN  650
C      *      W2(I)       : SECOND-ORDER I-TH FREQUENCY COMPONENT VALUE *AMN  660
C      *      Y2(P,I)     : PORT P SECOND-ORDER OUTPUT AT W2(I)        *AMN  670
C      *      FC2(I)      : COMBINATION CODE FOR W2(I)                 *AMN  680
C      *      W3(I)       : THIRD-ORDER I-TH FREQUENCY COMPONENT VALUE *AMN  690
C      *      Y3(P,I)     : PORT P THIRD-ORDER OUTPUT AT W3(I)        *AMN  700
C      *      FC3(I)      : FREQUENCY COMBINATION CODE FOR W3(I)       *AMN  710
C      *      FR(I)       : I-TH FREQUENCY VALUE IN THE COMPLETE SPECTRUM *AMN  720
C      *      Y(I)        : OUTPUT VOLTAGE AT FR(I)                   *AMN  730
C      *      IPT         : ARRAY USED FOR <HISTOGRAM> ANALYSIS         *AMN  740
C      *      YLG         : LOG OF THE OUTPUT (Y), USED FOR PLOTTING    *AMN  750
C      *      ST1,ST2     : DUMMY STORAGE ARRAYS USED FOR EQUIVALENCING *AMN  760
C      *      NCONT       : ARRAY FOR ADDRESSING NONLINEAR CURRENT     *AMN  770
C      *                  SOURCES AND REQUESTED OUTPUT PORTS (SEE      *AMN  780
C      *                  TECHNICAL REPORT FOR DETAILS)                *AMN  790
C      *      JCONT(K)    : SECOND CONTROLLING BRANCH NUMBER FOR NONLINEAR *AMN  800
C      *                  ELEMENT K/ SUBSEQUENTLY IDENTIFIES THE      *AMN  810
C      *                  NONLINEAR ELEMENT TYPE                      *AMN  820
C      *      TITLE      : ARRAY USED FOR READING TITLE AND OPTION CARD *AMN  830
C      *      NCAP        : NUMBER OF LINEAR CAPACITORS                *AMN  840
C      *      NDUS        : NUMBER OF LINEAR DEPENDENT VOLTAGE SOURCES *AMN  850
C      *      NRES        : NUMBER OF LINEAR RESISTORS                 *AMN  860
C      *      NIND        : NUMBER OF LINEAR INDUCTORS                 *AMN  870
C      *      NDCS        : NUMBER OF LINEAR DEPENDENT CURRENT SOURCES *AMN  880
C      *      NCS         : NUMBER OF LINEAR CURRENT SOURCES(=> OF PORTS) *AMN  890
C      *                  *AMN  900
C*****AMN  910
C      *                  *AMN  920
C      *                  *AMN  930
C      *                  *AMN  940
C      *                  *AMN  950
C      *                  *AMN  960
C      *                  *AMN  970
C      *                  *AMN  980
C      *                  *AMN  990
C***** ARRAYS REQUIRED FOR STORING AUGMENTED LINEAR NETWORK      *AMN 1000
C      *                  *AMN 1010
C      *      DIMENSION BR(75), NFROM(75), NTO(75), TYPE(75), ICONT(75), KEY(75) *AMN 1020
C      *                  *AMN 1030
C***** ARRAY FOR ELEMENT VALUES                                *AMN 1040
C      *                  *AMN 1050
C      *      DIMENSION VALUE(75)                                       *AMN 1060
C      *                  *AMN 1070
C***** ARRAYS FOR NONLINEAR ELEMENT TYPE AND POLYNOMIAL COEFFICIENTS *AMN 1080
C      *                  *AMN 1090
C      *      COMMON /001/ NTYPE(10),COFF(10,9)                        *AMN 1100
C      *                  *AMN 1110
C***** INPUT AMPLITUDE AND FREQUENCY ARRAYS                    *AMN 1120
C      *                  *AMN 1130
C      *      COMMON /003/ FREQ(10),AMP(10),TH(10),LUNIT              *AMN 1140
C      *      DIMENSION PHASE(5)                                       *AMN 1150
C      *                  *AMN 1160
C***** APPRAYS FOR HYBRID ANALYSIS                              *AMN 1170
C      *                  *AMN 1180
C      *      DIMENSION A(30,75), ANS(75,150), HEADER(300), EXTR1(75,30), EXTR2(175,30) *AMN 1190
C      *                  *AMN 1200

```

C***** ARRAYS FOR THE FORMATION OF STATE EQUATIONS	AMN 1210
C	AMN 1220
DIMENSION AMAT(20,20), BMAT(20,20), CMAT(25,20), DMAT(25,25)	AMN 1230
C	AMN 1240
C***** EIGENVALUE AND EIGENVECTOR ARRAYS	AMN 1250
C	AMN 1260
COMPLEX EVALS(20),EVECTS(20,20)	AMN 1270
C	AMN 1280
C***** ARRAYS FOR STORING PFE INFO AND WORK ARRAYS	AMN 1290
C	AMN 1300
COMPLEX DMAT(20,25),CHAT(25,20),WK1(20,20),WK2(20,20)	AMN 1310
C	AMN 1320
C***** ARRAYS FOR FREQUENCY SWEEP	AMN 1330
C	AMN 1340
COMMON /004/ NSTPS(5),FRINC(5),HFR(5)	AMN 1350
C	AMN 1360
C***** ARRAYS FOR FIRST-ORDER TRANSFER FUNCTIONS	AMN 1370
C	AMN 1380
COMPLEX Y1(25,10)	AMN 1390
C	AMN 1400
C***** ARRAYS FOR SECOND-ORDER TRANSFER FUNCTIONS	AMN 1410
C	AMN 1420
COMPLEX Y2(25,55),WKZ(25,25)	AMN 1430
DIMENSION I2(55)	AMN 1440
INTEGER FC2(55)	AMN 1450
C	AMN 1460
C***** ARRAYS FOR THIRD-ORDER TRANSFER FUNCTIONS	AMN 1470
C	AMN 1480
COMPLEX Y3(25,130)	AMN 1490
DIMENSION I3(130)	AMN 1500
INTEGER FC3(130)	AMN 1510
C	AMN 1520
C***** ARRAYS FOR COMPLETE OUTPUT SPECTRUM	AMN 1530
C	AMN 1540
COMPLEX Y(160)	AMN 1550
DIMENSION FR(160), IPT(160), YLG(160)	AMN 1560
C	AMN 1570
C***** MISCELLANEOUS WORK ARRAYS	AMN 1580
C	AMN 1590
DIMENSION ST1(75,100), ST2(50,225), NLBN(32), TITLE(80), NPORT(25)	AMN 1600
COMMON /016/ NCONT(22),JCONT(10)	AMN 1610
C	AMN 1620
COMMON /ETYPE/ R,G,L,C,E,IS,CU,UU,CC,UC	AMN 1630
COMMON /ENOS/ NCAP,INDUS,NRES,HIND,NDCS,NCS	AMN 1640
C	AMN 1650
C***** EQUIVALENCE FOR PHASE 1 (OBTAIN HYBRID MATRIX)	AMN 1660
C	AMN 1670
EQUIVALENCE (ST1(1),TITLE(1))	AMN 1680
EQUIVALENCE (ST1(1),DR(1))	AMN 1690
EQUIVALENCE (ST1(75),NFROM(1))	AMN 1700
EQUIVALENCE (ST1(151),NTO(1))	AMN 1710
EQUIVALENCE (ST1(225),TYPE(1))	AMN 1720
EQUIVALENCE (ST1(301),ICONT(1))	AMN 1730
EQUIVALENCE (ST1(375),VALUE(1))	AMN 1740
EQUIVALENCE (ST1(451),KEY(1))	AMN 1750
EQUIVALENCE (ST1(525),EXTR1(1))	AMN 1760
EQUIVALENCE (ST1(600),EXTR2(1))	AMN 1770
EQUIVALENCE (ST1(675),A(1))	AMN 1780
EQUIVALENCE (ST1(750),HEADER(1))	AMN 1790
EQUIVALENCE (ST2(1),ANS(1))	AMN 1800

```

C
C***** EQUIVALENCE FOR PHASE 2 (OBTAIN STATE EQUATIONS)
C
EQUIVALENCE (ST1(355),AMAT(1))
EQUIVALENCE (ST1(356),DMAT(1))
EQUIVALENCE (ST1(357),DMAT(1))
EQUIVALENCE (ST1(358),DMAT(1))
C
C***** EQUIVALENCE FOR PHASE 3 (OBTAIN Z(S) IN PFE FORM)
C
EQUIVALENCE (ST2(2001),UK1(1))
EQUIVALENCE (ST2(2002),UK2(1))
EQUIVALENCE (ST2(2003),EJECTS(1))
EQUIVALENCE (ST2(1),DMAT(1))
EQUIVALENCE (ST2(1001),DMAT(1))
C
C***** EQUIVALENCE FOR PHASE 4 (OBTAIN OUTPUT RESPONSES)
C
EQUIVALENCE (ST2(2004),Y1(1))
EQUIVALENCE (ST2(2005),Y2(1))
EQUIVALENCE (ST2(2006),Y3(1))
EQUIVALENCE (ST2(2007),Y4(1))
EQUIVALENCE (ST2(2008),Y5(1))
EQUIVALENCE (ST2(2009),Y6(1))
EQUIVALENCE (ST2(2010),Y7(1))
EQUIVALENCE (ST2(2011),Y8(1))
EQUIVALENCE (ST2(2012),Y9(1))
C
C***** EQUIVALENCE FOR PHASE 5 (OBTAIN COMPLETE OUTPUT SPECTRUM)
C
EQUIVALENCE (ST1(1),FR(1))
EQUIVALENCE (ST1(101),LFR(1))
EQUIVALENCE (ST1(102),LFR(1))
EQUIVALENCE (ST1(103),LFR(1))
EQUIVALENCE (ST1(104),LFR(1))
DATA C,L,E,IS/EN,CEN,L,EN,E,2H,I/
DATA R,G,UV,C,UC,UC/EN,R,2H,G,2HUV,2HCU,2HCC,2HUC/
DATA NC,ML,IR,ND/ENDC,2HML,2HMR,2HND/
C
C***** MAX CIRCUIT CONFIGURATION 75 30 NODES AND 75 BRANCHES
C
CALL SECOND (T0)
IRNODE=0
IRXNODE=30
IRBR=75
IRFRT=35
IRSTU=20
C
C***** MAX NONLINEAR ELEMENTS IS 10 (DEPENDENT TYPE LE 5)
C***** MAX INDEPENDENT SOURCES IS 2
C***** MAX TOTAL CAPACITORS AND INDUCTORS IS 20
C
NCAP=0
NRES=0
NIND=0
NCS=0
NCP=0
NCCS=0
NOUT=1
NOUT=2
C
C***** READ TITLE CARD

```

```

C      READ (5,212) (TITLE(I),J=1,30)
C
C***** WRITE TITLE LINE AND PRINT HEADING OF NETWORK DESCRIPTION
C
C      WRITE (6,214) (TITLE(I),J=1,30)
C
C***** READ DECODE AND WRITE DECODED OPTIONS
C
C      READ (5,206) (TITLE(1),J=1,10)
C      CALL DECODE (TITLE(1),F0,KK,SE,NM,PR,PC,AP)
C      WRITE (6,215)
C      WRITE (6,216)
C      WRITE (6,217)
C      WRITE (6,218)
C
C***** READ ANALYSIS PARAMETER CARD
C
C      READ (5,201) NLELEN,NLELEN,NFREQ,LUNIT,INTYP
C      NAF=NLELEN*NLELEN
C
C***** READ LINEAR CIRCUIT DESCRIPTION
C
C      DO 100 KK=1,NLELEN
C        READ (5,213) NEG,N1,N2,NT,UNL,ICT,OUTPT,NOT
C
C***** FORM LINEAR ELEMENT PROPERTY ARRAYS
C
C        KAPEN
C        KAPENK
C        KAPENKAPEN
C        KAPENKAPEN
C        KAPENKAPEN
C        KAPENKAPEN
C        KAPENKAPEN
C        KAPENKAPEN
C
C***** IF FREQUENT BRANCH IS AN OUTPUT,EXTRACT IT AS A PORT
C
C        KENUNUN
C        IF (OUTPT.EQ.0) GO TO 102
C        CALL OUTPUT (CARD,PR,TYPE,VALUE,NFROM,NTD,KEY,N1,N2,KEYU,IS,0.0,CAPEN
C
C          1  K
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C
C          102  KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C          KENUNUN
C
C***** SORT LINEAR ELEMENTS
C
C      IF (UNL.EQ.0) GO TO 111
C      IF (UNL.EQ.1) GO TO 112
C      IF (UNL.EQ.2) GO TO 113
C      IF (UNL.EQ.3) GO TO 114
C      IF (UNL.EQ.4) GO TO 115
C      IF (UNL.EQ.5) GO TO 116
C      IF (UNL.EQ.6) GO TO 117
C      IF (UNL.EQ.7) GO TO 118
C      IF (UNL.EQ.8) GO TO 119
C      IF (UNL.EQ.9) GO TO 120
C
C***** C
C

```

KEY(K)=8	AMN 3010
NDCS=NDCS+1	AMN 3020
GO TO 110	AMN 3030
C	AMN 3040
C***** U C U S	AMN 3050
C	AMN 3060
104 KEY(K)=3	AMN 3070
NDUS=NDUS+1	AMN 3080
GO TO 110	AMN 3090
C	AMN 3100
C***** C C U S	AMN 3110
C	AMN 3120
106 KEY(K)=4	AMN 3130
NDUS=NDUS+1	AMN 3140
GO TO 110	AMN 3150
C	AMN 3160
C***** U C C S	AMN 3170
C	AMN 3180
108 KEY(K)=7	AMN 3190
NDCS=NDCS+1	AMN 3200
C	AMN 3210
C***** WRITE DEPENDENT SOURCE BRANCH INFORMATION	AMN 3220
C	AMN 3230
110 WRITE (6,228) BR(K),NFROM(K),NTD(K),TYPE(K),VALUE(K),ICONT(K)	AMN 3240
GO TO 120	AMN 3250
C	AMN 3260
C***** RESISTIVE BRANCH	AMN 3270
C	AMN 3280
112 KEY(K)=5	AMN 3290
NRES=NRES+1	AMN 3300
GO TO 118	AMN 3310
C	AMN 3320
C***** CAPACITIVE BRANCH	AMN 3330
C	AMN 3340
114 NCAP=NCAP+1	AMN 3350
KEY(K)=2	AMN 3360
GO TO 118	AMN 3370
C	AMN 3380
C***** INDUCTIVE BRANCH	AMN 3390
C	AMN 3400
116 NIND=NIND+1	AMN 3410
KEY(K)=6	AMN 3420
C	AMN 3430
C***** WRITE R,L,C BRANCH INFORMATION	AMN 3440
C	AMN 3450
118 WRITE (6,230) BR(K),NFROM(K),NTD(K),TYPE(K),VALUE(K)	AMN 3460
120 CONTINUE	AMN 3470
C	AMN 3480
C***** READ NONLINEAR ELEMENT INFORMATION	AMN 3490
C	AMN 3500
KK=0	AMN 3510
DO 128 K=1,NNELEM	AMN 3520
KK=KK+1	AMN 3530
C	AMN 3540
C***** READ NONLINEAR ELEMENT TOPOLOGY	AMN 3550
C	AMN 3560
READ (5,232) NEG,N1,N2,NT,ICT,JCT	AMN 3570
IF (NT.EQ.2HTR) GO TO 126	AMN 3580
C	AMN 3590
C***** READ POLYNOMIAL COEFFICIENTS FOR THE NONLINEARITY	AMN 3600

C	IF (NT.EQ.ND) GO TO 122	ANN 3610
	READ (5,234) (COFF(KK,I),I=1,3)	ANN 3620
	GO TO 124	ANN 3630
122	READ (5,235) (COFF(KK,I),I=1,9)	ANN 3640
124	NLEN=KK+MIS	ANN 3650
	BRNEN=NEN	ANN 3660
	NFROM=NEN+1	ANN 3670
	NTO=NEN+12	ANN 3680
	TYPE=NEN+11	ANN 3690
	JCONT=NEN+10	ANN 3700
	JCONT=KK+207	ANN 3710
	GO TO 126	ANN 3720
126	CALL LTRANS (NEN,N1,NADD,KK,NLEN,DR,NFROM,NTD,TYPE,ICONT,VALUE,NNO	ANN 3730
	IDE=KEY)	ANN 3740
128	CONTINUE	ANN 3750
	NNLEN=KK	ANN 3760
C		ANN 3770
C****	WRITE NONLINEAR ELEMENT INFORMATION	ANN 3780
C		ANN 3790
	WRITE (5,252)	ANN 3800
	WRITE (5,253)	ANN 3810
	DO 127 I=1,NNLEN	ANN 3820
	IF (TYPE(N),EQ.ND) GO TO 130	ANN 3830
	WRITE (5,255) NFROM(I),NTD(I),TYPE(I),COFF(I,1),COFF(I,2),COFF(I,3)	ANN 3840
1	GO TO 132	ANN 3850
127	WRITE (5,256) NFROM(I),NTD(I),TYPE(I),ICONT(I),JCONT(I),COFF(I,4),	ANN 3860
1	COFF(I,5),COFF(I,6),COFF(I,7),COFF(I,8),COFF(I,9)	ANN 3870
	WRITE (5,257) COFF(I,4),COFF(I,5),COFF(I,6),COFF(I,7),COFF(I,8),COFF(I,9)	ANN 3880
122	CONTINUE	ANN 3890
C		ANN 3900
C****	READ AND WRITE GENERATOR INFORMATION	ANN 3910
C		ANN 3920
	READ (5,238) N1,N2,NT,ZS	ANN 3930
	IF (NT.EQ.R).OR.(NT.EQ.G)) GO TO 134	ANN 3940
	IF (NT.EQ.D) GO TO 135	ANN 3950
	CALL CXTPT (NADD,DR,TYPE,VALUE,NFROM,NTD,KEY,N1,N2,S,NT,ZS)	ANN 3960
	NEND=NEND+1	ANN 3970
	N2=N2	ANN 3980
	GO TO 133	ANN 3990
134	CALL CXTPT (NADD,DR,TYPE,VALUE,NFROM,NTD,KEY,N1,N2,S,NT,ZS)	ANN 4000
	NRES=NRES+1	ANN 4010
	N2=N2	ANN 4020
	GO TO 133	ANN 4030
135	CALL CXTPT (NADD,DR,TYPE,VALUE,NFROM,NTD,KEY,N1,N2,2,NT,ZS)	ANN 4040
	NCON=NCON+1	ANN 4050
	N2=N2	ANN 4060
136	CALL CXTPT (NADD,DR,TYPE,VALUE,NFROM,NTD,KEY,N1,N2,9,IS,0.00)	ANN 4070
	NCON=NCON+1	ANN 4080
	NCON=NCON+1	ANN 4090
	DO 140 I=1,NFROM	ANN 4100
140	READ (5,239) AMP(I),PHASE(I),FREQ(I),HFR(I),NSTPS(I)	ANN 4110
	WRITE (5,240)	ANN 4120
	WRITE (5,241) N1,N2,ZS,NT	ANN 4130
	WRITE (5,242) AMP(I)	ANN 4140
	DO 142 I=1,NFROM	ANN 4150
142	WRITE (5,243) I,FREQ(I),AMP(I),PHASE(I)	ANN 4160
		ANN 4170
		ANN 4180
		ANN 4190
		ANN 4200

IF (FS.NE.1) GO TO 145	AMN 4210
MXINC=0	AMN 4220
DO 144 I=1,NFREQ	AMN 4230
144 MXINC=MAX0(MXINC,NSTPS(I))	AMN 4240
WRITE (6,208) INTYP,MXINC	AMN 4250
146 IF (MX.NE.1) GO TO 154	AMN 4260
C	AMN 4270
C***** READ AND WRITE SECOND-GENERATOR INFORMATION	AMN 4280
C	AMN 4290
NFREQ=NFREQ+1	AMN 4300
READ (5,248) N1,N2,NT,ZS1,AMP(NFREQ),FREQ(NFREQ),PHASE(NFREQ)	AMN 4310
IF ((NT.EQ.R).OR.(NT.EQ.G)) GO TO 148	AMN 4320
IF (NT.EQ.C) GO TO 150	AMN 4330
CALL CXTPT (NADD,BR,TYPE,VALUE,NFROM,NT,KEY,N1,N2,S,NT,ZS1)	AMN 4340
NIND=NIND+1	AMN 4350
NZT1=3	AMN 4360
GO TO 152	AMN 4370
148 CALL CXTPT (NADD,BR,TYPE,VALUE,NFROM,NT,KEY,N1,N2,S,NT,ZS1)	AMN 4380
NRES=NRES+1	AMN 4390
NZT1=1	AMN 4400
GO TO 152	AMN 4410
150 CALL CXTPT (NADD,BR,TYPE,VALUE,NFROM,NT,KEY,N1,N2,2,NT,ZS1)	AMN 4420
NCAP=NCAP+1	AMN 4430
NZT1=2	AMN 4440
152 CALL CXTPT (NADD,BR,TYPE,VALUE,NFROM,NT,KEY,N1,N2,35,IS,0.0)	AMN 4450
NCS=NCS+1	AMN 4460
WRITE (6,242) N1,N2,ZS1,NT	AMN 4470
WRITE (6,244) LUNIT	AMN 4480
WRITE (6,246) NFREQ,FREQ(NFREQ),AMP(NFREQ),PHASE(NFREQ)	AMN 4490
C	AMN 4500
C*****FORM THE APPROPRIATE AUGMENTED LINEAR NETWORK	AMN 4510
C	AMN 4520
154 NCT=NOUT	AMN 4530
DO 156 K=1,NNELEM	AMN 4540
N=NLBN(K)	AMN 4550
ICON=ICONT(N)	AMN 4560
JCON=JCONT(K)	AMN 4570
NNODE=MAX0(NNODE,NFROM(N),NT(N))	AMN 4580
KEYU=KEYU+1	AMN 4590
NCT=NCT+1	AMN 4600
CALL CAGMNT (K,N,KEYU,NADD,ICON,JCON,NCT,BR,NFROM,NT,TYPE,ICON,N	AMN 4610
1 T,VALUE,KEY)	AMN 4620
156 CONTINUE	AMN 4630
IF (MX.NE.1) GO TO 158	AMN 4640
LOSRC=NCT+1	AMN 4650
NCONT(LOSRC)=LOSRC	AMN 4660
158 NELEM=NADD	AMN 4670
C	AMN 4680
C***** SORT ELEMENT DATA	AMN 4690
C	AMN 4700
NICS=NCS	AMN 4710
CALL CSORT (NELEM,BR,NFROM,NT,TYPE,ICONT,VALUE,KEY)	AMN 4720
C	AMN 4730
C***** COMBINE PORTS WHICH APPEAR ACROSS SAME NODE PAIR	AMN 4740
C	AMN 4750
CALL CRDPRT (NELEM,BR,NFROM,NT,KEY,TYPE,VALUE,ICONT)	AMN 4760
C	AMN 4770
C***** CONSECUTIVELY NUMBER THE EXTRACTED <INDEPENDENT> PORTS	AMN 4780
C	AMN 4790
DO 166 I=1,NICS	AMN 4800

IF (NCONT(I).EQ.I) GO TO 163	AMN 4810
J=I+1	AMN 4820
160 IF (NCONT(J).GT.I) GO TO 162	AMN 4830
IF (J.EQ.NICS) GO TO 165	AMN 4840
J=J+1	AMN 4850
GO TO 160	AMN 4860
162 DO 164 K=J,NICS	AMN 4870
164 IF (NCONT(K).EQ.J) NCONT(K)=I	AMN 4880
165 CONTINUE	AMN 4890
C	AMN 4900
C***** REMINDER THE CONTROLLING PORTS FOR THE NONLINEAR ELEMENTS	AMN 4910
C***** AND ASSIGN NUMERICAL IDENTIFIER(JCONT()) WITH EACH NONLINEAR	AMN 4920
C***** ELEMENT TYPE	AMN 4930
C	AMN 4940
DO 166 K=1,NICS	AMN 4950
166 NLEN(K)=NCONT(K)	AMN 4960
K1=NOUT	AMN 4970
J3=NOUT	AMN 4980
DO 168 K=1,NNELEM	AMN 4990
K1=K1+1	AMN 5000
J1=J3+1	AMN 5010
J2=J1+1	AMN 5020
J3=J2+1	AMN 5030
IF (NTYPE(K).EQ.NR) GO TO 174	AMN 5040
IF (NTYPE(K).EQ.ND) GO TO 170	AMN 5050
IF (NTYPE(K).EQ.NL) GO TO 172	AMN 5060
IF (NTYPE(K).EQ.NB) GO TO 173	AMN 5070
170 JCONT(K)=1	AMN 5080
GO TO 176	AMN 5090
172 JCONT(K)=2	AMN 5100
GO TO 176	AMN 5110
174 JCONT(K)=4	AMN 5120
176 NCDUM=NLEN(K1)	AMN 5130
NCONT(J1)=NCDUM	AMN 5140
NCONT(J2)=NCDUM	AMN 5150
NCONT(J3)=NCDUM	AMN 5160
GO TO 180	AMN 5170
173 NCONT(J1)=NLEN(K1)	AMN 5180
NCONT(J2)=NLEN(K1+1)	AMN 5190
K1=K1+2	AMN 5200
NCONT(J3)=NLEN(K1)	AMN 5210
JCONT(K)=3	AMN 5220
180 CONTINUE	AMN 5230
IF (NR.NE.1) GO TO 182	AMN 5240
LGSRG=NOUT+NNELEM+NNELEM+NNELEM+1	AMN 5250
NCONT(LGSRG)=NLEN(NICS)	AMN 5260
182 N37U=NCDUM+1	AMN 5270
NDR=NLEN	AMN 5280
C	AMN 5290
C*****PRINT AUGMENTED LINEAR NETWORK DESCRIPTION	AMN 5300
C	AMN 5310
WRITE (6,264)	AMN 5320
WRITE (6,265)	AMN 5330
WRITE (6,266)	AMN 5340
DO 184 K=1,NDR	AMN 5350
184 WRITE (6,269) DR(K),NFROM(K),NTO(K),TYPE(K),VALUE(K),ICONT(K)	AMN 5360
C	AMN 5370
C***** WRITE EXTRACTED PORT INFORMATION	AMN 5380
C	AMN 5390
WRITE (6,283)	AMN 5400

```

NDUM=NDR-NDS
DO 100 I=1,NDS
  NPORT(I)=0
  NFROM=NDUM+1
  NTO=NDUM
103 CONTINUE
C
C***** ZERO OUT A MATRIX
C
  DO 102 I=1,NMODE
    DO 100 J=1,NDR
      A(I,J)=0
100 CONTINUE
C
C***** STORE ENTRIES INTO A MATRIX
C
  DO 100 K=1,NLEN
    NFROM=NFROM(K)
    NTO=NTO(K)
    IF (NFROM.NE.0) A(NFROM,K)=1
    IF (NTO.NE.0) A(NTO,K)=-1
100 CONTINUE
  IF (DEBUG.1) GO TO 104
C
C***** PRINT THE A MATRIX FOR DEBUG RUN
C
  WRITE (6,272)
  DO 102 I=1,NMODE
    102 WRITE (6,274) (A(I,J),J=1,NDR)
C
C***** FORMULATE INVERTED EQUATIONS
C
  104 CALL FWARD (NDR,NMODE,DD,NPORT1,ANSOOL,II,DR,TYPE,ISCT,VALUE,A,
    1 1ENDER,ANS-EXTRA,EXTRA,NMODE,NDR)
C
C***** FORMULATE STATE EQUATIONS
C
  CALL STATE (NPORT1,ANSOOL,II,NDR,NSTU,SE,AMAT,DMAT,CMAT,DMAT,VALUE,
    1 1,ANS-NDR,NNSTU,NPORT)
C
C*****OBTAIN AND PRINT THE EIGENVALUES AND THE EIGENVECTORS OF THE
C*****MATRIX A
C
  CALL FVEU (CMAT,NSTU,NNSTU,EVALS,EVECTS,VALUE,ERR)
C
C***** PRINT EIGENVALUE AND EIGENVECTOR INFORMATION, IF DESIRED
C
  IF (DEBUG.1) GO TO 200
  WRITE (6,273)
  DO 103 I=1,NSTU
    103 WRITE (6,270) EVALS(I)
    WRITE (6,270)
    DO 100 J=1,NDR
      100 WRITE (6,271) (EVECTS(I,J),J=1,NSTU)
C
C
C*****OBTAINING AND PRINT THE INVERSE OF VCO AT THE EXTRACTED POINT
C
  200 CALL VCO (CMAT,EVALS,EVECTS,CMAT,DMAT,CMAT,DMAT,CMAT,DMAT,NPORT,
    1 1,ANS-NDR,NNSTU)
  CALL VCOIN (CMAT)

```

[illegible]

```

218 FORMAT (//,1X,15HLINEAR ELEMENTS)
220 FORMAT (1H0,6HBRANCH,4X,4HFROM,3X,2HTO,5X,7HELEMENT,3X,7HELEMENT,3X,1H,
1X,7HCONTROL,/,1X,6HNUMBER,4X,4HNODE,2X,4HNODE,5X,4HTYPE,5X,5HVALUE,
2,4X,6HBRANCH)
222 FORMAT (1H,6(1H.),4X,4(1H.),2X,4(1H.),4X,7(1H.),3X,7(1H.),3X,7(1H.),
1.))
224 FORMAT (2I2,11,2A3)
226 FORMAT (3I3,A2,E10.3,I3,11,A1)
228 FORMAT (2X,I3,6X,I3,3X,I3,7X,A2,4X,E10.3,3X,I3)
230 FORMAT (2X,I3,5X,I3,3X,I3,7X,A2,4X,E10.3)
232 FORMAT (3I3,A2,2I3)
234 FORMAT (3E10.3)
236 FORMAT (3E10.3,/,E10.3)
238 FORMAT (2I3,A2,E10.3)
240 FORMAT (//,1X,15HSOURCE INFORMATION:)
242 FORMAT (1H0,4HFROM,2X,I3,2X,2HTO,2X,I3,5X,9HIMPEDANCE,2X,E10.3,3X,
1A2)
244 FORMAT (1H0,9HFREQUENCY,5X,9HVALUE(,A3,1H),5X,9HAMPLITUDE,4X,10HPOWER,
1A5(,DEG),/,1H,9(1H.),5X,10(1H.),5X,9(1H.),4X,10(1H.))
246 FORMAT (1H,4X,I3,3X,E10.3,3X,E10.3,2X,E10.3)
248 FORMAT (2I3,A2,4E10.3)
250 FORMAT (4E10.3,I2)
252 FORMAT (//,1X,15HNONLINEAR ELEMENTS)
254 FORMAT (1H0,2X,4HFROM,5X,2HTO,5X,4HTYPE,4X,7HCONTROL,5X,22HPOLYNOMIAL,
11AL COEFFICIENTS,/,1H,2X,4HNODE,4X,4HNODE,11X,3H(1),5X,3H(2),/,1H,
22X,4(1H.),4X,4(1H.),5X,4(1H.),2X,11(1H.),3X,23(1H.))
256 FORMAT (1H0,3X,I2,6X,I2,7X,A2,16X,3HA1=,E12.4,3X,3HA2=,E12.4,3X,3HA3=,
1A3=,E12.4)
258 FORMAT (1H,38X,4HA02=,E12.4,2X,4HA11=,E12.4,2X,4HA03=,E12.4)
260 FORMAT (1H,23X,4HA03=,E12.4,2X,4HA21=,E12.4,2X,4HA12=,E12.4)
262 FORMAT (1H0,3X,I2,6X,I2,7X,A2,3X,I2,5X,I2,3X,4HA10=,E12.4,2X,4HA01=,
1=,E12.4,2X,4HA20=,E12.4)
264 FORMAT (//,1X,35HAugmented LINEAR NETWORK DESCRIPTION)
266 FORMAT (2X,I3,6X,I3,3X,I3,7X,A2,4X,E10.3,3X,I3,5X)
268 FORMAT (1H0,17HPORT ASSIGNMENTS:/,1H0,4X,4HPORT,4X,3HNODE PAIR,/,1H0,
1,3X,6HNUMBER,3X,4HFROM,2X,2HTO,/,1H,3X,5(1H.),3X,4(1H.),1X,4(1H.),
2)
270 FORMAT (6X,I2,5X,I2,3X,I2)
272 FORMAT (//,5H A MATRIX)
274 FORMAT (1H0,40I3)
276 FORMAT (/,12H EIGENVALUES)
278 FORMAT (/,2(3X,E12.4))
280 FORMAT (/,13H MODAL MATRIX)
282 FORMAT (/,4(2X,E12.3,3X,E12.3))
284 FORMAT (1H0,25HTIME FOR FORMING ZDC(SEC),F10.4)
286 FORMAT (1H,39HTIME FOR OBTAINING OUTPUT SPECTRUM(SEC),F10.4)
288 FORMAT (1H,25HTOTAL EXECUTION TIME(SEC),F10.4)
C
END
SUBROUTINE DOPRNS (TITLE,DB,FS,MX,SE,NM,PR,PC,AP)
C *****
C *
C ***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:
C * 1. SET THE FLAG VARIABLES FOR THE USER REQUESTED OPTIONS,
C * AND ALSO PRINT THESE OPTIONS.
C *
C ***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:
C * TITLE : USER REQUESTED OPTIONS
C * DB : FLAG VARIABLE FOR DEBUG RUN

```

```

C      *      FS      : FLAG VARIABLE FOR FREQUENCY SWEEP      *BOP  120
C      *      NK      : FLAG VARIABLE FOR TWO INPUT SOURCES    *EOP  130
C      *      SE      : FLAG VARIABLE FOR STATE EQUATION PRINT-OUT *BOP  140
C      *      NM      : FLAG VARIABLE FOR EIGENVALUE-EIGENVECTOR *EOP  150
C      *      : INFORMATION PRINT-OUT *EOP  160
C      *      PR      : FLAG VARIABLE FOR POLE-RESIDUE INFORMATION *EOP  170
C      *      : OF ZOC PRINT-OUT *EOP  180
C      *      PC      : FLAG VARIABLE FOR COMPLETE SPECTRUM PRINT *EOP  190
C      *      : AND PLOT *EOP  200
C      *      AP      : FLAG VARIABLE FOR <ALL> PORT PRINT-OUT *EOP  210
C      *      : *EOP  220
C*****EOP  230
C      INTEGER TITLE(1),DD,FS,SE,PR,PC,AP,LANS(2)
C      DATA LANS(1),LANS(2)/CHYES,CH NO/
C      :EOP  240
C***** INITIALIZE OPTION CONTROLLING FLAG VARIABLES
C      :EOP  250
C      DD=2
C      :EOP  260
C      FS=2
C      :EOP  270
C      NK=2
C      :EOP  280
C      SE=2
C      :EOP  290
C      NM=2
C      :EOP  300
C      PR=2
C      :EOP  310
C      PC=2
C      :EOP  320
C      AP=2
C      :EOP  330
C***** RESET FLAG VARIABLE VALUES FOR THE REQUESTED OPTIONS
C      :EOP  340
C      DO 100 I=1,3
C      :EOP  350
C      IFUM=TITLE(I)
C      :EOP  360
C      IF (IDUM.EQ.2H ) GO TO 102
C      :EOP  370
C      IF (IDUM.EQ.2HDD) GO TO 102
C      :EOP  380
C      GO TO 104
C      :EOP  390
C      102 DD=1
C      :EOP  400
C      GO TO 100
C      :EOP  410
C      104 IF (IDUM.EQ.2HFS) GO TO 105
C      :EOP  420
C      GO TO 103
C      :EOP  430
C      105 FS=1
C      :EOP  440
C      GO TO 100
C      :EOP  450
C      106 IF (IDUM.EQ.2HNK) GO TO 110
C      :EOP  460
C      GO TO 112
C      :EOP  470
C      110 NK=1
C      :EOP  480
C      GO TO 100
C      :EOP  490
C      112 IF (IDUM.EQ.2HSE) GO TO 114
C      :EOP  500
C      GO TO 116
C      :EOP  510
C      114 SE=1
C      :EOP  520
C      GO TO 100
C      :EOP  530
C      116 IF (IDUM.EQ.2HNM) GO TO 118
C      :EOP  540
C      GO TO 120
C      :EOP  550
C      118 NM=1
C      :EOP  560
C      GO TO 100
C      :EOP  570
C      120 IF (IDUM.EQ.2HPR) GO TO 122
C      :EOP  580
C      GO TO 124
C      :EOP  590
C      122 PR=1
C      :EOP  600
C      GO TO 100
C      :EOP  610
C      124 IF (IDUM.EQ.2HPC) GO TO 126
C      :EOP  620
C      GO TO 128
C      :EOP  630
C      126 PC=1
C      :EOP  640
C      GO TO 100
C      :EOP  650
C      :EOP  660
C      :EOP  670
C      :EOP  680
C      :EOP  690
C      :EOP  700
C      :EOP  710

```

128	IF (IDUM.EQ.2HAP) AP=1	DGP	720
130	CONTINUE	DGP	730
C		DGP	740
C*****	PRINT THE OPTIONS LIST	DGP	750
C		DGP	760
132	WRITE (6,124)	DGP	770
	WRITE (6,125) LANS(DB)	DGP	780
	WRITE (6,133) LANS(FS)	DGP	790
	WRITE (6,140) LANS(MX)	DGP	800
	WRITE (6,142) LANS(SE)	DGP	810
	WRITE (6,144) LANS(MM)	DGP	820
	WRITE (6,145) LANS(PR)	DGP	830
	WRITE (6,146) LANS(PC)	DGP	840
	WRITE (6,150) LANS(AP)	DGP	850
	RETURN	DGP	860
C		DGP	870
134	FORMAT (1H0,23HUSER REQUESTED OPTIONS:)	DGP	880
135	FORMAT (1H ,2X,16HDEBUG PRINT-OUT: ,1X,A3)	DGP	890
136	FORMAT (1H ,2X,27HFREQUENCY SWEEP CAPABILITY: ,1X,A3)	DGP	900
140	FORMAT (1H ,2X,18HIND-INPUT CIRCUIT: ,1X,A3)	DGP	910
142	FORMAT (1H ,2X,25HSTATE EQUATION PRINT-OUT: ,1X,A3)	DGP	920
144	FORMAT (1H ,2X,35HEIGENVALUES MODAL MATRIX PRINT-OUT: ,1X,A3)	DGP	930
146	FORMAT (1H ,2X,40HOPEN-CIRCUIT IMPEDANCE MATRIX PRINT-OUT: ,1X,A3)	DGP	940
148	FORMAT (1H ,2X,30HCOMPLETE OUTPUT SPECTRUM PLOT: ,1X,A3)	DGP	950
150	FORMAT (1H ,2X,27HALL EXTRACTED PORT OUTPUTS: ,1X,A3)	DGP	960
C		DGP	970
	END	DGP	980
	SUBROUTINE CASMNT (K,N,NKEY,NADD,ICON,JCON,NCT,BR,NFROM,NTD,TYPE,ICNS	DGP	990
	ICONT,VALUE,KEY)	DGP	1000
C		DGP	1010
C*****		DGP	1020
C	*	DGP	1030
C*****	THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	DGP	1040
C	*	DGP	1050
C	1. FORM THE AUGMENTED LINEAR NETWORK BY LUMPING THE	DGP	1060
C	LINEAR PART OF EACH NONLINEAR ELEMENT WITH THE	DGP	1070
C	EXISTING LINEAR NETWORK.	DGP	1080
C	*	DGP	1090
C*****	THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINE:	DGP	1100
C	1. CXTPT	DGP	1110
C	*	DGP	1120
C*****	THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	DGP	1130
C	K : NUMBER OF THE NONLINEAR ELEMENT	DGP	1140
C	N : USER SPECIFIED BRANCH NUMBER FOR THE K-TH	DGP	1150
C	NONLINEAR ELEMENT	DGP	1160
C	NKEY : KEY VALUE FOR THE NONLINEAR ELEMENT PORT BRANCH	DGP	1170
C	NADD : CURRENT HIGHEST BRANCH NUMBER IN THE LINEAR	DGP	1180
C	NETWORK	DGP	1190
C	ICON : NONLINEAR ELEMENT FIRST-CONTROLLING BRANCH NO.	DGP	1200
C	JCON : NONLINEAR ELEMENT SECOND-CONTROLLING BR NO.	DGP	1210
C	NCT : CURRENT NONLINEAR ELEMENT PORT NUMBER	DGP	1220
C	ARRAY NAMES AS DEFINED IN SUB-PROGRAM ANAETH	DGP	1230
C	*	DGP	1240
C*****		DGP	1250
C		DGP	1260
	INTEGER BR,TYPE,C,L,S,P,G,UN,NC,CN,CC	DGP	1270
	DIMENSION BR(1), NFROM(1), NTD(1), TYPE(1), ICONT(1)	DGP	1280
	DIMENSION VALUE(1)	DGP	1290
	DIMENSION KEY(1)	DGP	1300
	COMMON /001/ NTYPE(10),COFF(10,0)	DGP	1310
	COMMON /016/ ICONT(32),JCONT(32)	DGP	1320

COMMON /ENDS/ NCAP,NDUS,NRES,NIND,NDCS,NCS	CAG 340
DATA C,L,G,UC,IS/2H C,2H L,2H G,2H UC,2H I/	CAG 350
DATA NC,NL/2HNC,2HNL/	CAG 350
C	CAG 370
C*****APPLY A ZERO-VALUED CURRENT SOURCE ACROSS THE NONLINEAR ELEMENT	CAG 380
C	CAG 390
NTYPE(K)=TYPE(N)	CAG 400
TYPE(N)=IS	CAG 410
VALUE(N)=0.00	CAG 420
KEY(N)=IKEY	CAG 430
NCONT(NCT)=NCT	CAG 440
NCS=NCS+1	CAG 450
C	CAG 460
C*****CHECK FOR A DEPENDENT NONLINEAR ELEMENT	CAG 470
C	CAG 480
IF (ICON.LE.0) GO TO 104	CAG 490
NOT=NOT+1	CAG 500
IKEY=IKEY+1	CAG 510
CALL CXTERT (NADD,DR,TYPE,VALUE,NFROM,NTO,KEY,NFROM(ICON),NTO(ICON),	CAG 520
1,IKEY,IS,0.00)	CAG 530
NCS=NCS+1	CAG 540
VALUE(NADD)=0.000	CAG 550
KEY(NADD)=IKEY	CAG 560
NCONT(NCT)=NCT	CAG 570
IKEY=IKEY+1	CAG 580
NOT=NOT+1	CAG 590
CALL CXTERT (NADD,DR,TYPE,VALUE,NFROM,NTO,KEY,NFROM(JCON),NTO(JCON),	CAG 600
1,IKEY,IS,0.00)	CAG 610
NCS=NCS+1	CAG 620
NCONT(NCT)=NCT	CAG 630
C	CAG 640
C*****COMBINE THE LINEAR PART OF THE NONLINEARITY WITH THE LINEAR NTWK	CAG 650
C	CAG 660
IF (COFF(K,2).EQ.0.00) GO TO 102	CAG 670
CALL CXTERT (NADD,DR,TYPE,VALUE,NFROM,NTO,KEY,NFROM(N),NTO(N),8,UCCAG	CAG 680
1,COFF(K,2))	CAG 690
ICONT(NADD)=JCONT(K)	CAG 700
JCONT(K)=0	CAG 710
NDCS=NDCS+1	CAG 720
102 IF (COFF(K,1).EQ.0.00) RETURN	CAG 730
CALL CXTERT (NADD,DR,TYPE,VALUE,NFROM,NTO,KEY,NFROM(N),NTO(N),8,UCCAG	CAG 740
1,COFF(K,1))	CAG 750
ICONT(NADD)=ICONT(N)	CAG 760
ICONT(N)=0	CAG 770
NDCS=NDCS+1	CAG 780
RETURN	CAG 790
C	CAG 800
C*****INDEPENDENT TYPE NONLINEARITY	CAG 810
C	CAG 820
104 IF (COFF(K,1).EQ.0.00) RETURN	CAG 830
IF (NTYPE(K).EQ.NC) GO TO 105	CAG 840
IF (NTYPE(K).EQ.NL) GO TO 103	CAG 850
CALL CXTERT (NADD,DR,TYPE,VALUE,NFROM,NTO,KEY,NFROM(N),NTO(N),5,G,	CAG 860
1,COFF(K,1))	CAG 870
NRES=NRES+1	CAG 880
RETURN	CAG 890
105 CALL CXTERT (NADD,DR,TYPE,VALUE,NFROM,NTO,KEY,NFROM(N),NTO(N),2,C,	CAG 900
1,COFF(K,1))	CAG 910
NCAP=NCAP+1	CAG 920
RETURN	CAG 930

```

103 RECVL=1.0000/COFF(K,1)          CDS 540
CALL CWTPT (NADB,DR,TYPE,VALUE,NFROM,NTD,KEY,NFROM(N),NTD(N),S,L,CAS 550
1RECVL)                             CDS 560
NIND=NIND+1                         CDS 570
RETURN                              CDS 580
C                                  CDS 590
END                                  CDS 1000
SUBROUTINE CWTPT (NDR,DR,NFROM,NTD,KEY,TYPE,VALUE,ICONT) CDS 10
C                                  CDS 11
C***** CDS 12
C * CDS 13
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTIONS: CDS 14
C * 1. IDENTIFY ALL PARALLEL CAPACITOR, INDUCTOR, OR CURRENT CDS 15
C * SOURCE BRANCHES. CDS 16
C * 2. COMBINE THESE PARALLEL BRANCHES. CDS 17
C * CDS 18
C***** THIS SUB-PROGRAMS GLOSSARY OF PORTION NAMES: CDS 19
C * NDR : TOTAL NUMBER OF LINEAR ELEMENT BRANCHES CDS 20
C * ALL OTHER VARIABLE AND ARRAY NAMES AS DEFINED PREVIOUSLY CDS 21
C * IN SUB-PROGRAMS MAIN AND CASPT. CDS 22
C * CDS 23
C***** CDS 24
C CDS 25
C INTEGER DR(1),NFROM(1),NTD(1),KEY(1),TYPE(1),ICONT(1) CDS 26
DIMENSION VALUE(1) CDS 27
COMMON /019/ NDR(32),ICONT(32) CDS 28
COMMON /ENDS/ NCAP,NIND,NRES,NIDUS,NDCS,NDS CDS 29
ICONT=0 CDS 30
IFLAG=0 CDS 31
NDR=NCAP+NIND+NRES+NIDUS+NDCS CDS 32
C CDS 33
C***** IDENTIFY PARALLEL CAPACITOR, INDUCTOR, AND CURRENT SOURCE BRANCHES CDS 34
C CDS 35
102 IFLAG=IFLAG+1 CDS 36
GO TO (104,105,103), IFLAG CDS 37
104 N=NCAP CDS 38
NF=0 CDS 39
KS=1 CDS 40
GO TO 110 CDS 41
105 N=NIND+NRES+NIDUS CDS 42
KS=NF+NIDUS+NRES+1 CDS 43
GO TO 110 CDS 44
103 N=NDS+NDCS CDS 45
KS=NF+NDCS+1 CDS 46
110 IF (N.EQ.0) GO TO 102 CDS 47
NF=NF+1 CDS 48
I=KS CDS 49
112 IF (1.6T,NF) GO TO 125 CDS 50
IF (DR(I).EQ.0) GO TO 124 CDS 51
I1=I+1 CDS 52
DO 122 J=I1,NF CDS 53
IF ((NFROM(I).EQ.NFROM(J)).AND.(NTD(I).EQ.NTD(J)).AND.(TYPE(I). CDS 54
1 EQ.TYPE(J))) GO TO 111 CDS 55
GO TO 122 CDS 56
114 ICNT=ICNT+1 CDS 57
NDR=NDR-1 CDS 58
GO TO (116,110,120), IFLAG CDS 59
C CDS 60
C***** PARALLEL CAPACITOR CDS 61
C CDS 62

```

116	VALUE(I)=VALUE(I)+VALUE(J)	CRD	540
	NCON=NCON+1	CRD	550
	GO TO 102	CRD	560
C		CRD	570
C*****	PARALLEL INDUCTOR	CRD	580
C		CRD	590
118	VALUE(I)=VALUE(I)*VALUE(J)/(VALUE(I)+VALUE(J))	CRD	600
	NIND=NIND+1	CRD	610
	GO TO 112	CRD	620
120	ITIME=ITIME+1	CRD	630
	JTIME=JTIME+1	CRD	640
	ICONT(JTIME)=ICONT(JTIME)	CRD	650
	NDS=NDS+1	CRD	660
122	CONTINUE	CRD	670
124	I=I+1	CRD	680
	GO TO 112	CRD	690
126	IF (CPLD(1,1,0)) GO TO 102	CRD	700
	IF (CEND(1,0,0)) RETURN	CRD	710
C		CRD	720
C*****	REMOVE ALL PARALLEL BRANCHES IDENTIFIED PREVIOUSLY	CRD	730
C		CRD	740
	I=1	CRD	750
128	IF (CPLD(1,1,0)) GO TO 133	CRD	760
	IF (CEND(1,0,0)) GO TO 133	CRD	770
	GO TO 134	CRD	780
130	I=NDR+1	CRD	790
	GO TO 131	CRD	800
	DR(I)=DR(I)+1	CRD	810
	NRCH(I)=NRCH(I)+1	CRD	820
	NTD(I)=NTD(I)+1	CRD	830
	KEY(I)=KEY(I)+1	CRD	840
	VALUE(I)=VALUE(I)+1	CRD	850
	TYPE(I)=TYPE(I)+1	CRD	860
	ICONT(I)=ICONT(I)+1	CRD	870
132	CONTINUE	CRD	880
	NDR=NDR+1	CRD	890
	GO TO 128	CRD	900
134	I=I+1	CRD	910
	GO TO 128	CRD	920
136	RETURN	CRD	930
C		CRD	940
	END	CRD	950
	SUBROUTINE CSORT (NLEN, DR, NFROM, NTO, TYPE, ICONT, VALUE, KEY)	EST	10
C		EST	20
C*****		EST	30
C		EST	40
C*****	THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	EST	50
C	1. ARRANGE THE BRANCHES OF THE AUGMENTED LINEAR NETWORK	EST	60
C	IN AN ORDER SUITABLE FOR HYBRID AND STATE EQUATION	EST	70
C	FORMULATION.	EST	80
C		EST	90
C*****	THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	EST	100
C	NLEN : TOTAL NUMBER OF LINEAR ELEMENT BRANCHES	EST	110
C	NFROM : FROM NAMES AS DEFINED IN SUB-PROGRAM MAIN	EST	120
C		EST	130
C		EST	140
C		EST	150
C*****	PROGRAMME ELEMENT DATA TO ORDERING REQUIRED FOR HYBRID AND	EST	160
C*****	STATE SUBROUTINES VIA SHELL SORT METHOD	EST	170
C		EST	180

INTEGER BR,TYPE	CST	190
DIMENSION BR(1), NFROM(1), NTO(1), TYPE(1), ICONT(1)	CST	200
DIMENSION VALUE(1)	CST	210
DIMENSION KEY(1)	CST	220
COMMON /018/ NCONT(32),JCONT(10)	CST	230
J=1	CST	240
102 J=2*J	CST	250
I=NELEM/J	CST	260
IF (I.EQ.0) GO TO 103	CST	270
L=1	CST	280
M=I+1	CST	290
104 IF (M.GT.NELEM) GO TO 102	CST	300
LL=L	CST	310
MM=M	CST	320
IF (KEY(M).GE.KEY(L)) GO TO 103	CST	330
ITEMP=KEY(M)	CST	340
KEY(M)=KEY(L)	CST	350
KEY(L)=ITEMP	CST	360
TEMP=VALUE(M)	CST	370
VALUE(M)=VALUE(L)	CST	380
VALUE(L)=TEMP	CST	390
ITEMP=BR(M)	CST	400
BR(M)=BR(L)	CST	410
BR(L)=ITEMP	CST	420
ITEMP=NFROM(M)	CST	430
NFROM(M)=NFROM(L)	CST	440
NFROM(L)=ITEMP	CST	450
ITEMP=NTO(M)	CST	460
NTO(M)=NTO(L)	CST	470
NTO(L)=ITEMP	CST	480
ITEMP=TYPE(M)	CST	490
TYPE(M)=TYPE(L)	CST	500
TYPE(L)=ITEMP	CST	510
ITEMP=ICONT(M)	CST	520
ICONT(M)=ICONT(L)	CST	530
ICONT(L)=ITEMP	CST	540
L=L-1	CST	550
IF (L.LT.1) GO TO 103	CST	560
M=M-1	CST	570
GO TO 104	CST	580
106 L=LL+1	CST	590
M=MM+1	CST	600
GO TO 104	CST	610
108 RETURN	CST	620
C	CST	630
END	CST	640
SUBROUTINE EXTART (J,BR,TYPE,VALUE,NFROM,NTO,KEY,N1,N2,KEYU,NT,T)	CPT	10
C	CPT	20
C*****	CPT	30
C	CPT	40
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	CPT	50
C	CPT	60
C	CPT	70
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	CPT	80
C	CPT	90
C	CPT	100
C	CPT	110
C	CPT	120
C	CPT	130
C	CPT	140
C	CPT	150
C	CPT	160
C	CPT	170
C	CPT	180
C	CPT	190
C	CPT	200
C	CPT	210
C	CPT	220
C	CPT	230
C	CPT	240
C	CPT	250
C	CPT	260
C	CPT	270
C	CPT	280
C	CPT	290
C	CPT	300
C	CPT	310
C	CPT	320
C	CPT	330
C	CPT	340
C	CPT	350
C	CPT	360
C	CPT	370
C	CPT	380
C	CPT	390
C	CPT	400
C	CPT	410
C	CPT	420
C	CPT	430
C	CPT	440
C	CPT	450
C	CPT	460
C	CPT	470
C	CPT	480
C	CPT	490
C	CPT	500
C	CPT	510
C	CPT	520
C	CPT	530
C	CPT	540
C	CPT	550
C	CPT	560
C	CPT	570
C	CPT	580
C	CPT	590
C	CPT	600
C	CPT	610
C	CPT	620
C	CPT	630
C	CPT	640
C	CPT	650
C	CPT	660
C	CPT	670
C	CPT	680
C	CPT	690
C	CPT	700
C	CPT	710
C	CPT	720
C	CPT	730
C	CPT	740
C	CPT	750
C	CPT	760
C	CPT	770
C	CPT	780
C	CPT	790
C	CPT	800
C	CPT	810
C	CPT	820
C	CPT	830
C	CPT	840
C	CPT	850
C	CPT	860
C	CPT	870
C	CPT	880
C	CPT	890
C	CPT	900
C	CPT	910
C	CPT	920
C	CPT	930
C	CPT	940
C	CPT	950
C	CPT	960
C	CPT	970
C	CPT	980
C	CPT	990
C	CPT	1000

C	*	ARRAY NAMES AS DEFINED IN SUB-PROGRAM AMAIN	*CPT	150
C	*		*CPT	160
C	*****		*CPT	170
C			CPT	180
		INTEGER ER,TYPE,C,L,E,IS,R,G,UU,CU,CC,UC	CPT	190
		DIMENSION DR(1), NFROM(1), NTO(1), TYPE(1), VALUE(1), KEY(1)	CPT	200
C			CPT	210
C	*****	ADD A BRANCH	CPT	220
C			CPT	230
		J=J+1	CPT	240
		DR(J)=J	CPT	250
		TYPE(J)=NT	CPT	260
		VALUE(J)=T	CPT	270
		NFROM(J)=N1	CPT	280
		NTO(J)=N2	CPT	290
		KEY(J)=KEYU	CPT	300
		RETURN	CPT	310
C			CPT	320
		END	CPT	330

	SUBROUTINE DFTREE (NROW,NCOL,INDCOL,A,MU)	DTR	10
C		DTR	20
C	*****	DTR	30
C	*	*DTR	40
C	***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	*DTR	50
C	* 1. FIND THE PROPER TREE FROM THE INCIDENCE MATRIX.	*DTR	60
C	*	*DTR	70
C	***** THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINES:	*DTR	80
C	* 1. DIAECH	*DTR	90
C	*	*DTR	100
C	***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	*DTR	110
C	* NROW : NUMBER OF ROWS IN THE INCIDENCE (A) MATRIX	*DTR	120
C	* NCOL : NUMBER OF COLUMNS IN THE INCIDENCE (A) MATRIX	*DTR	130
C	* INDCOL : INDEPENDENT COLUMNS OF THE A MATRIX	*DTR	140
C	*	*DTR	150
C	*****	DTR	160
C		DTR	170
C	SUBROUTINE DFTREE TAKES THE MATRIX A, APPLIES SUBROUTINE DIAECH	DTR	180
C	AND FINDS THE INDEPENDENT COLUMNS IN A CLOSEST TO THE LEFT.	DTR	190
C	THESE INDEPENDENT COLUMNS MAKE UP THE TREE BRANCHES.	DTR	200
C		DTR	210
	INTEGER A, INDCOL(NROW), COL, TEMP	DTR	220
	DIMENSION A(MU,1)	DTR	230
	L=1	DTR	240
	TEMP=1	DTR	250
	CALL DIAECH (NROW,NCOL,A,MU)	DTR	260
C		DTR	270
C	STEP THROUGH ROWS	DTR	280
C		DTR	290
	DO 104 K=1,NROW	DTR	300
C		DTR	310
C	STEP THROUGH COLUMNS	DTR	320
C		DTR	330
	DO 102 J=TEMP,NCOL	DTR	340
C		DTR	350
	FIND INDEPENDENT COLUMNS	DTR	360
C		DTR	370
	TEST IF ELEMENT EQUAL TO ONE	DTR	380
C		DTR	390
	IF (A(K,J).NE.1) GO TO 102	DTR	400
C		DTR	410
	RECORD INDEPENDENT COLUMN NUMBER	DTR	420
C		DTR	430
	INDCOL(L)=J	DTR	440
	L=L+1	DTR	450
	TEMP=J+1	DTR	460
	GO TO 104	DTR	470
	102 CONTINUE	DTR	480
	104 CONTINUE	DTR	490
	RETURN	DTR	500
C		DTR	510
	END	DTR	520
	SUBROUTINE DHYBRD (NBR,NNODE,DEBUG,NPORT1,ANSCOL,II,BR,TYPE,ICONT,	DND	10
	IVALUE,A,HEADER,ANS,F3,F6,MU,ME)	DND	20
C		DND	30
C	*****	DND	40
C	*	*DND	50
C	***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTIONS:	*DND	60
C	* 1. PERFORM HYBRID ANALYSIS OF THE AUGMENTED LINEAR CKT.	*DND	70
C	* 2. ACT AS THE EXECUTIVE CALLING PROGRAM FOR PERFORMING	*DND	80

C	*	THE HYBRID ANALYSIS.	*DHD	90
C	*		*DHD	100
C	*****	THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINES:	*DHD	110
C	*	1. DFTREE	*DHD	120
C	*	2. DIAECH	*DHD	130
C	*	3. DPRINT	*DHD	140
C	*	4. DRAECH	*DHD	150
C	*	5. DPRNT1	*DHD	160
C	*		*DHD	170
C	*****	THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	*DHD	180
C	*	NBR : NUMBER OF AUGMENTED LINEAR NETWORK BRANCHES	*DHD	190
C	*	NNODE : NUMBER OF INDEPENDENT NODES	*DHD	200
C	*	DEBUG : DEBUG OPTION FLAG VARIABLE	*DHD	210
C	*	NPORT1 : ADDRESS FOR LOCATING FIRST COLUMN OF MATRIX A	*DHD	220
C	*	IN THE HYBRID MATRIX	*DHD	230
C	*	ANSCOL : ADDRESS FOR LOCATING FIRST COLUMN OF MATRIX B	*DHD	240
C	*	II : ADDRESS FOR LOCATING FIRST ROW OF MATRIX A	*DHD	250
C	*	ALL OTHER VARIABLE NAMES AND ARRAYS AS DEFINED IN SUB-	*DHD	260
C	*	PROGRAM AMAIN	*DHD	270
C	*		*DHD	280
C	*****		*DHD	290
C			DHD	300
C		INTEGER A,TYPE,ICONT,UH,CH,DCOL(75),ICOUNT(2),COUN,BEGIN,TEMP,ST,TDHD	DHD	310
C		IN,TP,PORT,HEADER,BR,RBR(75),ANSROW,ANSCOL,DEBUG,ISTP	DHD	320
C		INTEGER R,G,C,E,CU,UU,CC,UC	DHD	330
C		DIMENSION BR(1),TYPE(1),ICONT(1)	DHD	340
C		DIMENSION VALUE(1)	DHD	350
C		DIMENSION A(MU,1),HEADER(300)	DHD	360
C		DIMENSION ANS(ME,1)	DHD	370
C		DIMENSION F3(ME,1),FS(ME,1)	DHD	380
C		COMMON /ETYPE/ R,G,L,C,E,IS,CU,UU,CC,UC	DHD	390
C		DATA CH,UH/1HI,1HU/	DHD	400
C		DO 102 I=1,NBR	DHD	410
C		DCOL(I)=0	DHD	420
C	102	RBR(I)=0	DHD	430
C			DHD	440
C		DETERMINE ELEMENTS MAKING UP THE TREE	DHD	450
C		CALL DFTREE (NNODE,NBR,DCOL,A,MU)	DHD	460
C			DHD	470
C		REORDER A MATRIX INTO FOUR CLASSES	DHD	480
C			DHD	490
C		1. TREE PORT BRANCHES (TP)	DHD	500
C		2. TREE NON-PORT BRANCHES (TN)	DHD	510
C		3. LINK NON-PORT BRANCHES (LN)	DHD	520
C		4. LINK PORT BRANCHES (LP)	DHD	530
C			DHD	540
C			DHD	550
C		DCOL CONTAINS ORDERING OF A WITH TREE BRANCHES IN LEFTMOST	DHD	560
C		COLUMNS	DHD	570
C			DHD	580
C		JJ=NNODE+1	DHD	590
C		N=1	DHD	600
C		DO 106 J=1,NNODE	DHD	610
C		M=DCOL(J)	DHD	620
C		DO 104 K=N,M	DHD	630
C		IF (M.EQ.K) GO TO 106	DHD	640
C		DCOL(JJ)=K	DHD	650
C		JJ=JJ+1	DHD	660
C			DHD	670
C			DHD	680

DO 108 I=N,NBR	DHD 690
108 DCOL(I)=I	DHD 700
C REORDER DCOL INTO FOUR CLASSES	DHD 710
C ICOUNT(1) MARKS LAST PORT COLUMN OF TREE BRANCHES	DHD 720
C ICOUNT(2) MARKS LAST NON-PORT COLUMN OF LINK BRANCHES	DHD 730
C	DHD 740
ICOUNT(1)=1	DHD 750
IT2=NNODE	DHD 760
I=1	DHD 770
110 DO 112 M=1,IT2	DHD 780
MM=(NBR+1)*(I-1)+((3-(2*I))*M)	DHD 790
ITEM=DCOL(MM)	DHD 800
IF (TYPE(ITEM).NE.S.AND.TYPE(ITEM).NE.C.AND.TYPE(ITEM).NE.L.AND	DHD 810
1 .TYPE(ITEM).NE.IS) GO TO 112	DHD 820
ITEM1=ICOUNT(I)	DHD 830
DCOL(MM)=DCOL(ITEM1)	DHD 840
DCOL(ITEM1)=ITEM	DHD 850
ICOUNT(I)=ICOUNT(I)+1-((I-1)*2)	DHD 860
112 CONTINUE	DHD 870
IF (I.EQ.2) GO TO 114	DHD 880
ICOUNT(1)=ICOUNT(1)-1	DHD 890
ICOUNT(2)=NBR	DHD 900
IT2=NBR-NNODE	DHD 910
I=2	DHD 920
GO TO 110	DHD 930
C	DHD 940
C REORDER THE A MATRIX AND THE ORIGINAL LABEL VECTOR TO	DHD 950
C CORRESPOND TO THE REORDERED DCOL	DHD 960
C	DHD 970
C	DHD 980
114 NN=2	DHD 990
N=1	DHD 1000
BEGIN=1	DHD 1010
COUN=0	DHD 1020
116 ITEM=DCOL(N)	DHD 1030
IF (ITEM.EQ.BEGIN) GO TO 120	DHD 1040
ITEMP=ER(N)	DHD 1050
ER(N)=ER(ITEM)	DHD 1060
ER(ITEM)=ITEMP	DHD 1070
DO 118 J=1,NNODE	DHD 1080
TEMP=A(J,N)	DHD 1090
A(J,N)=A(J,ITEM)	DHD 1100
118 A(J,ITEM)=TEMP	DHD 1110
COUN=COUN+1	DHD 1120
DCOL(N)=DCOL(N)	DHD 1130
N=ITEM	DHD 1140
GO TO 116	DHD 1150
120 DCOL(N)=DCOL(N)	DHD 1160
IF (COUN.EQ.(NBR-1)) GO TO 125	DHD 1170
DO 124 I=NN,NBR	DHD 1180
ITEM=DCOL(I)	DHD 1190
IF (ITEM.EQ.I) GO TO 122	DHD 1200
IF (ITEM.LT.0) GO TO 124	DHD 1210
BEGIN=I	DHD 1220
I=I	DHD 1230
GO TO 116	DHD 1240
122 COUN=COUN+1	DHD 1250
DCOL(I)=DCOL(I)	DHD 1260
NN=I	DHD 1270
124 CONTINUE	DHD 1280

125	DO 125 N=1,NDR	DHD 1290
126	DCOL(N)=IABS(DCOL(N))	DHD 1300
C		DHD 1310
C	REDUCE REORDERED A MATRIX TO ROW ECHELON FORM	DHD 1320
C		DHD 1330
C	CALL DRECH (NMODE,NDR,A,MU)	DHD 1340
C		DHD 1350
C	BACK SUBSTITUTE A MATRIX	DHD 1360
C		DHD 1370
	DO 130 I=2,NMODE	DHD 1380
	LRON=I-1	DHD 1390
	DO 130 J=1,LRON	DHD 1400
	IFCOL=I	DHD 1410
	ITEMP=A(J,IFCOL)	DHD 1420
	DO 130 K=I,NDR	DHD 1430
130	A(J,K)=A(J,K)-A(I,K)*ITEMP	DHD 1440
C		DHD 1450
C	FORMULATE THE ELEMENT CHARACTERISTICS	DHD 1460
C		DHD 1470
C	TP IS THE NUMBER OF COLUMNS IN F1 AND F5	DHD 1480
C	TH IS THE NUMBER OF COLUMNS IN F2 AND F3	DHD 1490
C	LI IS THE NUMBER OF COLUMNS IN F3 AND F7	DHD 1500
C	LP IS THE NUMBER OF COLUMNS IN F4 AND F8	DHD 1510
C		DHD 1520
	TP=ICOUNT(1)	DHD 1530
	TH=NMODE-ICOUNT(1)	DHD 1540
	LI=ICOUNT(2)-NMODE	DHD 1550
	LP=NDR-ICOUNT(2)	DHD 1560
	PORT=TP+LP	DHD 1570
	NPORT=TH+LI	DHD 1580
	ANSROW=NDR	DHD 1590
	ANSCOL=NDR+PORT	DHD 1600
	WRITE (6,272)	DHD 1610
	IF (TH.EQ.0) GO TO 132	DHD 1620
	IF (DEBUG.NE.1) GO TO 132	DHD 1630
	WRITE (6,274) (BR(I),I=1,TP)	DHD 1640
132	J=TP+1	DHD 1650
	IF (TH.EQ.0) GO TO 134	DHD 1660
	IF (DEBUG.NE.1) GO TO 134	DHD 1670
	WRITE (6,275) (BR(I),I=J,NMODE)	DHD 1680
134	J=NMODE+1	DHD 1690
	J=NMODE+LI	DHD 1700
	IF (LI.EQ.0) GO TO 133	DHD 1710
	IF (DEBUG.NE.1) GO TO 133	DHD 1720
	WRITE (6,273) (DR(I),I=J,JJ)	DHD 1730
133	J=JJ+1	DHD 1740
	IF (LP.EQ.0) GO TO 133	DHD 1750
	IF (DEBUG.NE.1) GO TO 133	DHD 1760
	WRITE (6,280) (DR(I),I=J,NDR)	DHD 1770
C		DHD 1780
C	ZERO GMS MATRIX	DHD 1790
C		DHD 1800
133	DO 140 I=1,ANSROW	DHD 1810
	DO 140 J=1,ANSCOL	DHD 1820
140	ANS(I,J)=0.0	DHD 1830
	DO 144 I=1,NPORT	DHD 1840
	DO 142 J=1,TP	DHD 1850
142	FG(I,J)=0.0	DHD 1860
	DO 144 J=1,LI	DHD 1870
144	FG(I,J)=0.0	DHD 1880

KOUNT=ICOUNT(1)	DHD 1880
K=0	DHD 1900
J=1	DHD 1910
DO 146 I=1,NBR	DHD 1920
ITEM=BR(I)	DHD 1930
146 RBR(ITEM)=I	DHD 1940
IF (DEBUG.NE.1) GO TO 148	DHD 1950
WRITE (6,282) TP,TN,LN,LP	DHD 1960
WRITE (6,284) (BR(I),I=1,NBR)	DHD 1970
148 KOUNT=KOUNT+1	DHD 1980
MM=DCOL(KOUNT)	DHD 1990
ITEMP=ICONT(MM)	DHD 2000
ITEMP=RBR(ITEMP)	DHD 2010
IT1=PORT+J	DHD 2020
IF (TYPE(MM).EQ.G.OR.TYPE(MM).EQ.UC.OR.TYPE(MM).EQ.CC) GO TO 152	DHD 2030
VOLTAGE SOURCE TYPE	DHD 2040
IF (KOUNT.GT.NNODE) GO TO 150	DHD 2050
F2	DHD 2060
IT2=LN+J	DHD 2070
ANS(IT1,IT2)=1.	DHD 2080
IF (TYPE(MM).EQ.CV) GO TO 158	DHD 2090
IF (TYPE(MM).EQ.UU) GO TO 165	DHD 2100
F6(J,J)=-VALUE(MM)	DHD 2110
GO TO 156	DHD 2120
150 K=K+1	DHD 2130
F3(J,K)=1.	DHD 2140
IF (TYPE(MM).EQ.CV) GO TO 158	DHD 2150
IF (TYPE(MM).EQ.UU) GO TO 165	DHD 2160
F7	DHD 2170
ANS(IT1,K)=-VALUE(MM)	DHD 2180
GO TO 156	DHD 2190
CURRENT SOURCE TYPE	DHD 2200
152 IF (KOUNT.GT.NNODE) GO TO 154	DHD 2210
F6(J,J)=1.	DHD 2220
IF (TYPE(MM).EQ.UC) GO TO 166	DHD 2230
IF (TYPE(MM).EQ.CC) GO TO 158	DHD 2240
F2	DHD 2250
IT2=LN+J	DHD 2260
ANS(IT1,IT2)=-VALUE(MM)	DHD 2270
GO TO 156	DHD 2280
154 K=K+1	DHD 2290
F7	DHD 2300
ANS(IT1,K)=1.	DHD 2310
IF (TYPE(MM).EQ.UC) GO TO 166	DHD 2320
IF (TYPE(MM).EQ.CC) GO TO 158	DHD 2330
F3(J,K)=-VALUE(MM)	DHD 2340
J=J+1	DHD 2350
156 IF (KOUNT.NE.ICOUNT(2)) GO TO 148	DHD 2360

GO TO 174	DHD 2490
C	DHD 2500
C	DHD 2510
C	DHD 2520
158 IF (ITEMP.GT.TP) GO TO 160	DHD 2530
C	DHD 2540
C	DHD 2550
C	DHD 2560
F5	DHD 2570
IT2=NPORT+ITEMP	DHD 2580
ANS(IT1,IT2)=-VALUE(MM)	DHD 2590
GO TO 153	DHD 2600
160 IF (ITEMP.GT.NNODE) GO TO 162	DHD 2610
IT=ITEMP-TP	DHD 2620
F6(J,IT)=-VALUE(MM)	DHD 2630
GO TO 153	DHD 2640
162 IF (ITEMP.GT.ICOUNT(2)) GO TO 164	DHD 2650
IT=ITEMP-NNODE	DHD 2660
C	DHD 2670
C	DHD 2680
C	DHD 2690
F7	DHD 2700
ANS(IT1,IT)=-VALUE(MM)	DHD 2710
GO TO 153	DHD 2720
164 IT=ITEMP-ICOUNT(2)	DHD 2730
C	DHD 2740
C	DHD 2750
C	DHD 2760
F8	DHD 2770
IT2=NDR+TP+IT	DHD 2780
ANS(IT1,IT2)=-VALUE(MM)	DHD 2790
GO TO 153	DHD 2800
C	DHD 2810
C	DHD 2820
C	DHD 2830
VOLTAGE CONTROLLED	DHD 2840
166 IF (ITEMP.GT.TP) GO TO 168	DHD 2850
C	DHD 2860
C	DHD 2870
C	DHD 2880
F1	DHD 2890
IT2=NDR+ITEMP	DHD 2900
ANS(IT1,IT2)=-VALUE(MM)	DHD 2910
GO TO 153	DHD 2920
168 IF (ITEMP.GT.NNODE) GO TO 170	DHD 2930
IT=ITEMP-TP	DHD 2940
C	DHD 2950
C	DHD 2960
C	DHD 2970
F2	DHD 2980
IT2=LH+IT	DHD 2990
ANS(IT1,IT2)=-VALUE(MM)	DHD 3000
GO TO 153	DHD 3010
170 IF (ITEMP.GT.ICOUNT(2)) GO TO 172	DHD 3020
IT=ITEMP-NNODE	DHD 3030
F3(J,IT)=-VALUE(MM)	DHD 3040
GO TO 153	DHD 3050
172 IT=ITEMP-ICOUNT(2)	DHD 3060
C	DHD 3070
C	DHD 3080
C	
F4	
IT2=NPORT+TP+IT	
ANS(IT1,IT2)=-VALUE(MM)	
GO TO 153	
174 IF (DEDUC.NE.1) GO TO 196	
IF (LH.EQ.0) GO TO 184	

C		DMD 3090
C	WRITE F3 FOR DEBUG RUN	DMD 3100
C		DMD 3110
	WRITE (6,285)	DMD 3120
	IT1=1	DMD 3130
176	IT2=LN	DMD 3140
	IF ((IT2-IT1).GT.10) GO TO 180	DMD 3150
	IF (IT2.EQ.IT1) GO TO 184	DMD 3160
	WRITE (6,288)	DMD 3170
	DO 178 I=1,NPORT	DMD 3180
178	WRITE (6,290) (F3(I,J),J=IT1,IT2)	DMD 3190
	GO TO 184	DMD 3200
180	IT2=IT1+9	DMD 3210
	WRITE (6,288)	DMD 3220
	DO 182 I=1,NPORT	DMD 3230
182	WRITE (6,290) (F3(I,J),J=IT1,IT2)	DMD 3240
	IT1=IT2+1	DMD 3250
	GO TO 176	DMD 3260
184	IF (TP.EQ.0) GO TO 194	DMD 3270
C		DMD 3280
C	WRITE F6 FOR DEBUG RUN	DMD 3290
C		DMD 3300
	WRITE (6,292)	DMD 3310
	IT1=1	DMD 3320
186	IT2=TN	DMD 3330
	IF ((IT2-IT1).GT.10) GO TO 190	DMD 3340
	IF (IT2.EQ.IT1) GO TO 194	DMD 3350
	WRITE (6,293)	DMD 3360
	DO 188 I=1,NPORT	DMD 3370
188	WRITE (6,290) (F6(I,J),J=IT1,IT2)	DMD 3380
	GO TO 194	DMD 3390
190	IT2=IT1+9	DMD 3400
	WRITE (6,293)	DMD 3410
	DO 192 I=1,NPORT	DMD 3420
192	WRITE (6,290) (F6(I,J),J=IT1,IT2)	DMD 3430
	IT1=IT2+1	DMD 3440
	GO TO 186	DMD 3450
194	WRITE (6,294)	DMD 3460
	CALL DPPRINT (ANSCL,ANSROW,ANS,ME)	DMD 3470
C		DMD 3480
C	ZEPO OUT F6	DMD 3490
C		DMD 3500
196	IF (TN.EQ.0) GO TO 205	DMD 3510
	DO 204 J=1,TN	DMD 3520
	KK=TP+J	DMD 3530
	DO 204 I=1,NPORT	DMD 3540
	IT1=PORT+I	DMD 3550
	IF (LN.EQ.0) GO TO 200	DMD 3560
C		DMD 3570
C	CHANGE F7	DMD 3580
C		DMD 3590
	DO 198 K=1,LN	DMD 3600
	LK=NNODE+K	DMD 3610
198	ANS(IT1,K)=ANS(IT1,K)-(F6(I,J)*FLOAT(A(KK,LK)))	DMD 3620
200	IF (LP.EQ.0) GO TO 204	DMD 3630
C		DMD 3640
C	CHANGE F8	DMD 3650
C		DMD 3660
	DO 202 K=1,LP	DMD 3670
	LK=ICOUNT(2)+K	DMD 3680

	IT2=NDR+TP*K	DHD 3690
202	ANS(IT1,IT2)=ANS(IT1,IT2)-(F5(I,J)*FLOAT(A(KK,LK)))	DHD 3700
204	CONTINUE	DHD 3710
C		DHD 3720
C	ZERO OUT F3	DHD 3730
C		DHD 3740
205	IF (LN.EQ.0) GO TO 215	DHD 3750
	DO 214 J=1,LN	DHD 3760
	KK=MNODE+J	DHD 3770
	DO 214 I=1,NPORT	DHD 3780
	IT1=PORT+I	DHD 3790
	IF (TI1.EQ.0) GO TO 210	DHD 3800
C		DHD 3810
C	CHANGE F2	DHD 3820
C		DHD 3830
	DO 203 K=1,TN	DHD 3840
	KK=TP*K	DHD 3850
	IT2=LN*K	DHD 3860
203	ANS(IT1,IT2)=ANS(IT1,IT2)-(F3(I,J)*FLOAT(-A(KK,LK)))	DHD 3870
210	IF (TP.EQ.0) GO TO 214	DHD 3880
C		DHD 3890
C	CHANGE F1	DHD 3900
C		DHD 3910
	DO 212 K=1,TP	DHD 3920
	IT2=NDR*K	DHD 3930
212	ANS(IT1,IT2)=ANS(IT1,IT2)-(F3(I,J)*FLOAT(-A(K,LK)))	DHD 3940
214	CONTINUE	DHD 3950
C		DHD 3960
C	FILL ANS MATRIX	DHD 3970
C		DHD 3980
215	IF (DEBVS.ME.1) GO TO 213	DHD 3990
	WRITE (S,225)	DHD 4000
	CALL DPRINT (ANSCOL,ANSROW,ANS,ME)	DHD 4010
213	IF (LN.EQ.0.OR.TP.EQ.0) GO TO 222	DHD 4020
C		DHD 4030
C	STORE D1	DHD 4040
C		DHD 4050
	DO 220 I=1,TP	DHD 4060
	DO 220 J=1,LN	DHD 4070
	KK=MNODE+J	DHD 4080
220	ANS(I,J)=A(I,K)	DHD 4090
222	LC=LN+1	DHD 4100
	ITEMP=TP+1	DHD 4110
	IF (ITEMP.GT.PORT.OR.LC.GT.NPORT) GO TO 226	DHD 4120
C		DHD 4130
C	STORE -D4 TRANSPOSE	DHD 4140
C		DHD 4150
	DO 224 I=ITEMP,PORT	DHD 4160
	JJ=LC+I-ITEMP*MNODE	DHD 4170
	DO 224 J=LC,NPORT	DHD 4180
	II=J+1-LC*TP	DHD 4190
224	ANS(I,J)=A(II,JJ)	DHD 4200
226	IF (TP.EQ.0) GO TO 230	DHD 4210
C		DHD 4220
C	STORE UNIT MATRIX ABOVE F5	DHD 4230
C		DHD 4240
	DO 223 I=1,TP	DHD 4250
	LD=NPORT+I	DHD 4260
223	ANS(I,LD)=1.0	DHD 4270
230	IF (LP.EQ.0) GO TO 234	DHD 4280

C		DHD 4290
C	STORE UNIT MATRIX ABOVE F4	DHD 4300
C		DHD 4310
	II=TP+1	DHD 4320
	DO 232 I=II,PORT	DHD 4330
	LD=NPORT+I	DHD 4340
232	ANS(I,LD)=1.0	DHD 4350
234	ITEMP=TP+1	DHD 4360
	LF=LD+TP	DHD 4370
	LE=LD+1	DHD 4380
	IF (ITEMP.GT.PORT.OR.LE.GT.LF) GO TO 238	DHD 4390
C		DHD 4400
C	STORE -D2 TRANSPOSE	DHD 4410
C		DHD 4420
	DO 236 I=ITEMP,PORT	DHD 4430
	JJ=I-ITEMP+ICOUNT(2)+1	DHD 4440
	DO 236 J=LE,LF	DHD 4450
	II=J+1-LE	DHD 4460
236	ANS(I,J)=-A(II,JJ)	DHD 4470
238	LE=LF+LP	DHD 4480
	LD=LF+1	DHD 4490
	IF (TP.EQ.0.OR.LD.GT.LE) GO TO 242	DHD 4500
C		DHD 4510
C	STORE D2	DHD 4520
C		DHD 4530
	DO 240 I=1,TP	DHD 4540
	DO 240 J=LD,LE	DHD 4550
	K=ICOUNT(2)+1+J-LD	DHD 4560
240	ANS(I,J)=A(I,K)	DHD 4570
242	IF (DEBUG.NE.1) GO TO 244	DHD 4580
	WRITE (6,298)	DHD 4590
	CALL DPRINT (ANSCOL,ANSROW,ANS,ME)	DHD 4600
C		DHD 4610
C	REDUCE ANS MATRIX TO ECHELON FORM	DHD 4620
C		DHD 4630
244	CALL DRAECH (NBR,ANSCOL,ANSROW,1,1,ANS,MU,ME)	DHD 4640
	ZERO=1.0000E-15	DHD 4650
	IF (DEBUG.NE.1) GO TO 246	DHD 4660
	WRITE (6,300)	DHD 4670
	CALL DPRINT (ANSCOL,ANSROW,ANS,ME)	DHD 4680
246	DO 248 I=1,NBR	DHD 4690
	DO 248 J=1,NPORT	DHD 4700
	II=NBR+1-I	DHD 4710
	IF (ABS(ANS(II,J)).LE.ZERO) ANS(II,J)=0.0	DHD 4720
	IF (ANS(II,J).NE.0.) GO TO 250	DHD 4730
248	CONTINUE	DHD 4740
250	II=II+1	DHD 4750
C		DHD 4760
C	FILL COLUMN HEADING VECTOR FOR FINAL DPRINT OUT	DHD 4770
C		DHD 4780
	J=0	DHD 4790
	IF (TP.EQ.0) GO TO 254	DHD 4800
	DO 252 I=1,TP	DHD 4810
	IT=2*I	DHD 4820
	HEADER(IT)=BR(I)	DHD 4830
	HEADER(IT-1)=CH	DHD 4840
	I2=2*(PORT+1)	DHD 4850
	HEADER(I2)=BR(I)	DHD 4860
252	HEADER(I2-1)=UH	DHD 4870
254	IF (LP.EQ.0) GO TO 258	DHD 4880

J=TP	DHD 4890
DO 256 I=1,LP	DHD 4900
J=J+1	DHD 4910
K=I+ICOUNT(2)	DHD 4920
IT=2*J	DHD 4930
HEADER(IT)=BR(K)	DHD 4940
HEADER(IT-1)=UH	DHD 4950
I2=2*(PORT+TP+I)	DHD 4960
HEADER(I2)=BR(K)	DHD 4970
255 HEADER(I2-1)=CH	DHD 4980
258 IT=4*PORT	DHD 4990
NPORT1=NPORT+1	DHD 5000
DO 260 I=II,NBR	DHD 5010
DO 260 J=NPORT1,ANSCOL	DHD 5020
260 IF (ABS(ANS(I,J)).LE.ZERO) ANS(I,J)=0.0	DHD 5030
IF (DEBUG.NE.1) GO TO 262	DHD 5040
C	DHD 5050
C	DHD 5060
C	DHD 5070
DPRINT FINAL ANS MATRIX FOR DEBUG RUN	DHD 5080
CALL DPRINT1 (IT,NPORT1,ANSCOL,II,NBR,HEADER,ANS,ME)	DHD 5090
262 IF (II.EQ.NBR) GO TO 265	DHD 5100
C	DHD 5110
C	DHD 5120
C	DHD 5130
BACK SUBSTITUTE FINAL ANSWER MATRIX	DHD 5140
IT1=ANSROW-II+1	DHD 5150
IT2=II+1	DHD 5160
DO 264 I=IT2,ANSROW	DHD 5170
C	DHD 5180
C	DHD 5190
C	DHD 5200
ANS(IRW,ICL) IS PIVOT ELEMENT USED TO ZERO ELEMENTS ABOVE	DHD 5210
IRW=ANSROW+IT2-I	DHD 5220
ICL=NPORT+IT1+IT2-I	DHD 5230
IT3=IRW-1	DHD 5240
C	DHD 5250
C	DHD 5260
C	DHD 5270
J=ROW ZEROING OUT ABOVE PIVOT	DHD 5280
DO 264 J=II,IT3	DHD 5290
B=ANS(J,ICL)	DHD 5300
C	DHD 5310
C	DHD 5320
C	DHD 5330
K=COLUMN CHANGING OF JTH ROW	DHD 5340
DO 264 K=ICL,ANSCOL	DHD 5350
264 ANS(J,K)=ANS(J,K)-B*ANS(IRW,K)	DHD 5360
265 DO 268 I=II,NBR	DHD 5370
DO 268 J=NPORT1,ANSCOL	DHD 5380
268 IF (ABS(ANS(I,J)).LE.ZERO) ANS(I,J)=0.0	DHD 5390
C	DHD 5400
C	DHD 5410
C	DHD 5420
DPRINT FINAL ANS MATRIX	DHD 5430
IF (DEBUG.NE.1) GO TO 270	DHD 5440
CALL DPRINT1 (IT,NPORT1,ANSCOL,II,NBR,HEADER,ANS,ME)	DHD 5450
270 RETURN	DHD 5460
C	DHD 5470
272 FORMAT (1H0,1CHTREE PORT BRANCHES,/30(1X,I2))	DHD 5480
274 FORMAT (1H0,1CHTREE NON-PORT BRANCHES,/30(1X,I2))	
276 FORMAT (1H0,2CHTREE NON-PORT BRANCHES,/30(1X,I2))	
278 FORMAT (1H0,2CHTREE NON-PORT BRANCHES,/30(1X,I2))	
280 FORMAT (1H0,1CHTREE PORT BRANCHES,/30(1X,I2))	
282 FORMAT (1H0, SHTP = ,I3/, 6H TN = ,I3/, 6H LN = ,I3/, 6H LP = ,I3/,	
1,I3)	

284	FORMAT (1H0, 2H2R,40(1X,12))	END 5480
285	FORMAT (///, 18H F3 BEFORE ZEROING)	END 5500
288	FORMAT (1X)	END 5510
290	FORMAT (1X,10(511.4,1X))	END 5520
292	FORMAT (///, 18H F5 BEFORE ZEROING)	END 5530
294	FORMAT (///, 25H AHS MATRIX BEFORE ZEROING)	END 5540
295	FORMAT (///, 25H AHS MATRIX AFTER ZEROING)	END 5550
298	FORMAT (///, 25H AHS MATRIX WITH D VALUES FILLED IN)	END 5560
300	FORMAT (///, 35H AHS MATRIX REDUCED TO ECHELON FORM)	END 5570
C	END	END 5580
	SUBROUTINE DIAECH (NROW,NCOL,A,MV)	END 5590
C		END 5600
C	*****	END 5610
C	*	END 5620
C	***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	END 5630
C	* 1. MANIPULATE THE INCIDENCE (A) MATRIX INTO ECHELON FORM	END 5640
C	*	END 5650
C	***** THIS SUB-PROGRAMS GLOSSARY OF FORTRAN NAMES:	END 5660
C	* NROW : NUMBER OF ROWS IN THE A MATRIX	END 5670
C	* NCOL : NUMBER OF COLUMNS IN THE A MATRIX	END 5680
C	*	END 5690
C	*****	END 5700
C		END 5710
C	SUBROUTINE DIAECH MANIPULATES MATRIX A INTO ECHELON FORM	END 5720
C		END 5730
C	INTEGER A,C,G,GPLUS1,P,B	END 5740
C	DIMENSION A(MV,1)	END 5750
C	C=1	END 5760
C	G=1	END 5770
102	DO 116 I=G,NROW	END 5780
	IF (A(I,C).EQ.0) GO TO 115	END 5790
C		END 5800
C	INTERCHANGE I AND G ROW TO GET NONZERO PIVOT	END 5810
C		END 5820
C	IF (I.EQ.G) GO TO 103	END 5830
	DO 104 K=C,NCOL	END 5840
	D=A(I,K)	END 5850
	A(I,K)=A(G,K)	END 5860
	A(G,K)=D	END 5870
104	CONTINUE	END 5880
C		END 5890
C	NORMALIZE ROW TO GET POSITIVE NUMBER FOR PIVOT	END 5900
C		END 5910
106	IF (A(G,C).GT.0) GO TO 110	END 5920
	DO 108 K=C,NCOL	END 5930
108	A(G,K)=-A(G,K)	END 5940
110	IF (G.GE.NROW) RETURN	END 5950
C		END 5960
C	ZERO COLUMN BELOW PIVOT	END 5970
C		END 5980
	GPLUS1=G+1	END 5990
	DO 114 P=GPLUS1,NROW	END 6000
	D=A(P,C)	END 6010
	IF (D.EQ.0) GO TO 114	END 6020
	DO 112 K=C,NCOL	END 6030
112	A(P,K)=-D*A(G,K)+A(P,K)	END 6040
114	CONTINUE	END 6050
	G=G+1	END 6060
	C=C+1	END 6070

GO TO 102	DTH	500
116 CONTINUE	DTH	510
IF (G.GT.NROW) RETURN	DTH	520
C=C+1	DTH	530
GO TO 102	DTH	540
C	DTH	550
END	DTH	560
SUBROUTINE DPRINT (ANSCOL,ANSROW,ANS,ME)	DPT	10
C	DPT	20
C*****	DPT	30
C	*DPT	40
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	*DPT	50
C	*DPT	60
C	*DPT	70
C	*DPT	80
C***** THIS SUB-PROGRAMS GLOSSARY OF FORTRAN VARIABLES:	*DPT	90
C	*DPT	100
C	*DPT	110
C*****	*DPT	120
C	DPT	130
C	DPT	140
C	DPT	150
C	DPT	160
C	DPT	170
C	DPT	180
C	DPT	190
C	DPT	200
102 IT2=ANSCOL	DPT	210
IF ((IT2-IT1).GT.9) GO TO 105	DPT	220
IF (IT2.EQ.IT1) RETURN	DPT	230
C	DPT	240
C	DPT	250
C	DPT	260
C	DPT	270
C	DPT	280
104 WRITE (6,112) (ANS(I,J),J=IT1,IT2)	DPT	290
RETURN	DPT	300
105 IT2=IT1+9	DPT	310
C	DPT	320
C	DPT	330
C	DPT	340
C	DPT	350
C	DPT	360
C	DPT	370
108 WRITE (6,112) (ANS(I,J),J=IT1,IT2)	DPT	380
IT1=IT2+1	DPT	390
GO TO 102	DPT	400
C	DPT	410
110 FORMAT (1X)	DPT	420
112 FORMAT (1X,10(E11.4,1X))	DPT	430
C	DPT	440
END	DPT	450
SUBROUTINE DPRINT1 (NDR,ACL1,ACL2,ARW1,ARW2,HEADER,ANS,ME)	DPT	10
C	DPT	20
C*****	DPT	30
C	*DPT	40
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	*DPT	50
C	*DPT	60
C	*DPT	70
C	*DPT	80
C***** THIS SUB-PROGRAMS GLOSSARY OF FORTRAN NAMES:	*DPT	90
C	*DPT	100
C	*DPT	110
C*****	*DPT	120
C	DPT	130
C	DPT	140
C	DPT	150
C	DPT	160
C	DPT	170
C	DPT	180
C	DPT	190
C	DPT	200
102 IT2=ANSCOL	DPT	210
IF ((IT2-IT1).GT.9) GO TO 105	DPT	220
IF (IT2.EQ.IT1) RETURN	DPT	230
C	DPT	240
C	DPT	250
C	DPT	260
C	DPT	270
C	DPT	280
104 WRITE (6,112) (ANS(I,J),J=IT1,IT2)	DPT	290
RETURN	DPT	300
105 IT2=IT1+9	DPT	310
C	DPT	320
C	DPT	330
C	DPT	340
C	DPT	350
C	DPT	360
C	DPT	370
108 WRITE (6,112) (ANS(I,J),J=IT1,IT2)	DPT	380
IT1=IT2+1	DPT	390
GO TO 102	DPT	400
C	DPT	410
110 FORMAT (1X)	DPT	420
112 FORMAT (1X,10(E11.4,1X))	DPT	430
C	DPT	440
END	DPT	450

```

C      *      ACL1      : FIRST COLUMN OF THE DESIRED PART      *DP1  100
C      *      ACL2      : LAST COLUMN OF THE DESIRED PART      *DP1  110
C      *      ARW1      : FIRST ROW OF THE DESIRED PART      *DP1  120
C      *      ARW2      : LAST ROW OF THE DESIRED PART      *DP1  130
C      *      HEADER     : COLUMN HEADING VECTOR      *DP1  140
C      *      ANS        : HYBRID MATRIX      *DP1  150
C      *      ME         : ROW DIMENSION OF ANS IN THE CALLING PROGRAM *DP1  160
C      *                                     *DP1  170
C*****DP1  180
C      SUBROUTINE DPRNT1 PRINTS ONLY THE DESIRED PART OF THE ANS      DP1  190
C      MATRIX DESCRIBING THE PORT EQUATIONS ALONG WITH THE COLUMN      DP1  200
C      HEADINGS      DP1  210
C      DP1  220
C      DP1  230
C      INTEGER A,HEADER,ACL1,ACL2,ARW1,ARW2,HDR      DP1  240
C      DIMENSION HEADER(300)      DP1  250
C      DIMENSION ANS(ME,1)      DP1  260
C      ITM2=ACL1-1      DP1  270
C      IT1=1      DP1  280
C 102 IT2=HDR      DP1  290
C      IF ((IT2-IT1).GT.19) GO TO 105      DP1  300
C      IF (IT2.EQ.IT1) RETURN      DP1  310
C      DP1  320
C      LESS OR EQUAL 10 COLUMNS TO PRINT      DP1  330
C      DP1  340
C      WRITE (6,110) (HEADER(I),I=IT1,IT2)      DP1  350
C      ITM1=ITM2+1      DP1  360
C      DO 104 I=ARW1,ARW2      DP1  370
C 104 WRITE (6,112) (ANS(I,J),J=ITM1,ACL2)      DP1  380
C      RETURN      DP1  390
C 106 IT2=IT1+19      DP1  400
C      DP1  410
C      MORE THAN 10 COLUMNS TO PRINT      DP1  420
C      DP1  430
C      WRITE (6,110) (HEADER(I),I=IT1,IT2)      DP1  440
C      ITM1=ITM2+1      DP1  450
C      ITM2=ITM1+9      DP1  460
C      DO 108 I=ARW1,ARW2      DP1  470
C 108 WRITE (6,112) (ANS(I,J),J=ITM1,ITM2)      DP1  480
C      IT1=IT2+1      DP1  490
C      GO TO 102      DP1  500
C      DP1  510
C 110 FORMAT (1H0,10(4X,A1,I2,5X))      DP1  520
C 112 FORMAT (1H0,10(21.4,1X))      DP1  530
C      DP1  540
C      END      DP1  550
C      SUBROUTINE DRAECH (N,N,MARK,ROW1,COL1,AD,MU,ME)      DRH  10
C      DRH  20
C*****DRH  30
C      *      *DRH  40
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:      *DRH  50
C      *      1. OPERATE ON THE ROWS OF THE HYBRID MATRIX TO REDUCE IT      *DRH  60
C      *      TO ECHELON FORM      *DRH  70
C      *      *DRH  80
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:      *DRH  90
C      *      N      : LAST ROW NUMBER IN ECHELON PART OF HYBRID MATRIX      *DRH  100
C      *      N      : LAST ROW NUMBER FOR ROW OPERATION      *DRH  110
C      *      MARK    : LAST COLUMN NUMBER IN ECHELON FORM MATRIX      *DRH  120
C      *      ROW1    : FIRST ROW NUMBER IN ECHELON FORM MATRIX      *DRH  130
C      *      COL1    : FIRST COLUMN NUMBER IN ECHELON FORM MATRIX      *DRH  140

```

C	*	ALL OTHER VARIABLE NAMES AS DEFINED IN SUB-PROGRAM AMAIN	*DRH	150
C	*		*DRH	160
C	*****		*****DRH	170
C		DIMENSION AD(ME,1)	DRH	180
		INTEGER C,G,GPLUS1,P,ROW1,COL1	DRH	190
C			DRH	200
C		DRARECH PERFORMS ROW OPERATIONS ON A TO REDUCE A TO ECHELON FORM	DRH	210
C			DRH	220
C		COLUMNS COL1 TO MARK ARE REDUCED TO ROW ECHELON FORM WHILE THE	DRH	230
C		ROW OPERATIONS ARE CARRIED OUT ON THE ROWS FROM MARK + 1 TO N.	DRH	240
C		ROWS ROW1 TO N ARE REDUCED TO ROW ECHELON FORM	DRH	250
C		G IS THE ROW IN WHICH WE ARE DETERMINING THE PIVOT POINT	DRH	260
C		C IS THE COLUMN IN WHICH WE ARE DETERMINING THE PIVOT POINT	DRH	270
C			DRH	280
		C=COL1-1	DRH	290
		G=ROW1	DRH	300
102		IF (C.EQ.MARK) RETURN	DRH	310
		C=C+1	DRH	320
C			DRH	330
C		FIND THE MAX NONZERO ELEMENT IN THE C COLUMN BELOW AND	DRH	340
C		INCLUDING PIVOT	DRH	350
C			DRH	360
		I=0	DRH	370
		ZERO=1.000E-15	DRH	380
		THZ=0.0	DRH	390
		DO 104 J=G,N	DRH	400
		IF (ABS(AD(J,C)).LE.ZERO) AD(J,C)=0.0	DRH	410
		IF (ABS(AD(J,C)).LE.THZ) GO TO 104	DRH	420
		THZ=ABS(AD(J,C))	DRH	430
		I=J	DRH	440
104		CONTINUE	DRH	450
		IF (THZ.EQ.0.0) GO TO 102	DRH	460
C			DRH	470
C		IF THE NONZERO ELEMENT IS IN THE PIVOT ROW, DO NOT EXCHANGE	DRH	480
C		ROWS	DRH	490
C			DRH	500
		IF (I.EQ.G) GO TO 103	DRH	510
C			DRH	520
C		EXCHANGE PIVOT ROW WITH ROW HAVING NONZERO ELEMENT IN PIVOT	DRH	530
C		COLUMN	DRH	540
C			DRH	550
		DO 103 K=C,N	DRH	560
		D=AD(I,K)	DRH	570
		AD(I,K)=AD(G,K)	DRH	580
103		AD(G,K)=D	DRH	590
C			DRH	600
C		CHECK IF PIVOT POINT ALREADY NORMALIZED TO 1	DRH	610
C			DRH	620
102		IF (AD(G,C).EQ.1.) GO TO 112	DRH	630
C			DRH	640
C		NORMALIZE PIVOT ROW	DRH	650
C			DRH	660
		ALPHA=AD(G,C)	DRH	670
		DO 110 L=G,N	DRH	680
		AD(G,K)=AD(G,K)/ALPHA	DRH	690
110		IF (ABS(AD(G,K)).LE.ZERO) AD(G,K)=0.0	DRH	700
C			DRH	710
C		CHECK IF JUST NORMALIZED PIVOT IN LAST ROW	DRH	720
C			DRH	730
			DRH	740

112 IF (G.GE.M) RETURN	DRH	750
C	DRH	760
C ZERO THE ELEMENTS BELOW THE PIVOT	DRH	770
C	DRH	780
GPLUS1=G+1	DRH	790
DO 116 P=GPLUS1,M	DRH	800
B=AD(P,C)	DRH	810
IF (ABS(AD(P,C)).LE.ZERO) AD(P,C)=0.0	DRH	820
IF (ABS(AD(P,C)).EQ.0.0) GO TO 116	DRH	830
DO 114 K=C,N	DRH	840
114 AD(P,K)=-B*AD(G,K)+AD(P,K)	DRH	850
116 CONTINUE	DRH	860
IF (G.GE.M) RETURN	DRH	870
G=G+1	DRH	880
GO TO 102	DRH	890
C	DRH	900
END	DRH	910
SUBROUTINE ESTATE (NPORT1,ANSCOL,II,NBR,NSU,DEBUG,A,B,C,D,VALUE,ANEST		10
1S,ME,NS,MP)	EST	20
C	EST	30
C*****	EST	40
C *	EST	50
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTIONS:	EST	60
C *	EST	70
C 1. OBTAIN THE MATRICES IN THE STATE SPACE REPRESENTATION	EST	80
C FOR THE AUGMENTED LINEAR NETWORK.	EST	90
C *	EST	100
C 2. PRINT THE STATE SPACE DESCRIPTION, IF REQUESTED.	EST	110
C ***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	EST	120
C *	EST	130
C NPORT1 : ADDRESS FOR LOCATING FIRST COLUMN OF MATRIX A	EST	140
C *	EST	150
C ANSCOL : ADDRESS FOR LOCATING FIRST COLUMN OF MATRIX B	EST	160
C *	EST	170
C II : ADDRESS FOR LOCATING FIRST ROW OF MATRIX A	EST	180
C *	EST	190
C NBR : TOTAL NUMBER OF BRANCHES IN LINEAR CIRCUIT	EST	200
C *	EST	210
C NSU : TOTAL NUMBER OF STATE VARIABLES	EST	220
C *	EST	230
C DEBUG : FLAG VARIABLE FOR PRINTING STATE EQUATIONS	EST	240
C *	EST	250
C A : MATRIX A IN STATE SPACE DESCRIPTION	EST	260
C *	EST	270
C B : MATRIX B IN STATE SPACE DESCRIPTION	EST	280
C *	EST	290
C C : MATRIX C IN STATE SPACE DESCRIPTION	EST	300
C *	EST	310
C D : MATRIX D IN STATE SPACE DESCRIPTION	EST	320
C *	EST	330
C VALUE : ARRAY OF ELEMENT VALUES	EST	340
C *	EST	350
C ANS : HYBRID MATRIX	EST	360
C *****	EST	370
C	EST	380
INTEGER ANSCOL,CONN,DEBUG	EST	390
DIMENSION VALUE(1)	EST	400
DIMENSION ANS(ME,1)	EST	410
DIMENSION A(NS,1), B(NS,1), C(MP,1), D(MP,1)	EST	420
DIMENSION DENOM(20)	EST	430
COMMON /ENOS/ NCAP,NDUS,NRES,NIND,NDCS,NCS	EST	440
NCPI=NCAP+1	EST	450
IF (NCAP.EQ.0) GO TO 104	EST	460
DO 102 I=1,NCAP	EST	470
102 DENOM(I)=VALUE(I)	EST	480
104 IF (NIND.EQ.0) GO TO 108	EST	490
K=NCAP+NDUS+NRES	EST	500
DO 106 I=NCPI,NSU	EST	510
K=K+1	EST	520
106 DENOM(I)=VALUE(K)	EST	530
108 NEONS=NBR+1-II	EST	540
C	EST	550

AD-A088 422

PURDUE UNIV LAFAYETTE IND
PRANC: PROGRAM FOR ANALYZING NONLINEAR CIRCUITS.(U)
MAY 80 H K THAPAR, B J LEON

F/6 9/3

F30602-78-C-0102

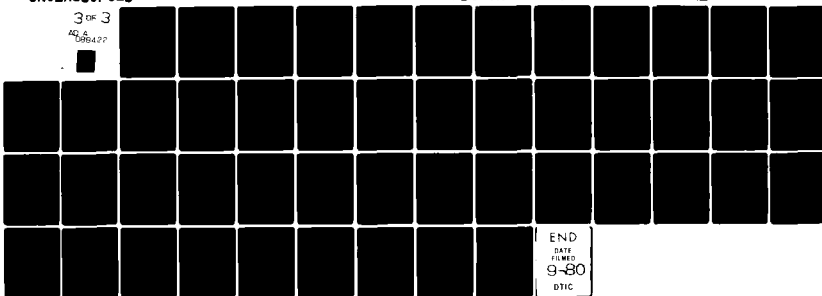
UNCLASSIFIED

RADC-TR-80-139

NL

3 OF 3

AD A088422



END
DATE
FILMED
9-80
DTIC

C	FILL MATRIX A	EST 440
C	N1=I1	EST 450
	N2=N1+NCA*+NIND-1	EST 460
	N3=NPORT1+NEONS	EST 470
	N4=N3+NCA*+NIND-1	EST 480
	IF (N7.LT.N1) GO TO 123	EST 490
	I1=0	EST 500
	DO 110 I=N1,N2	EST 510
	I1=I1+1	EST 520
	J1=0	EST 530
	DO 110 J=N3,N4	EST 540
	J1=J1+1	EST 550
	110 A(I1,J1)=-ANS(I,J)/DENOM(I1)	EST 560
C		EST 570
C	FILL MATRIX B	EST 580
C	NS=ANSCOL-NCS+1	EST 590
	NS=ANSCOL	EST 600
	I1=0	EST 610
	DO 114 I=N1,N2	EST 620
	I1=I1+1	EST 630
	J1=0	EST 640
	DO 112 J=NS,NS	EST 650
	J1=J1+1	EST 660
	112 B(I1,J1)=-ANS(I,J)/DENOM(I1)	EST 670
	114 CONTINUE	EST 680
C		EST 690
C	*****FILL MATRIX C	EST 700
C	I1=0	EST 710
	N1=N2+1	EST 720
	N2=N2+NCS	EST 730
	N3=ANSCOL-NCS-NSU+1	EST 740
	N4=N3+NSU	EST 750
	DO 116 I=N1,N2	EST 760
	I1=I1+1	EST 770
	J1=0	EST 780
	DO 116 J=N3,N4	EST 790
	J1=J1+1	EST 800
	C(I1,J1)=-ANS(I,J)	EST 810
	116 CONTINUE	EST 820
C		EST 830
C	*****FILL MATRIX D	EST 840
C	NS=ANSCOL-NCS+1	EST 850
	NS=NS+NCS	EST 860
	I1=0	EST 870
	DO 118 I=N1,N2	EST 880
	I1=I1+1	EST 890
	J1=0	EST 900
	DO 118 J=NS,NS	EST 910
	J1=J1+1	EST 920
	D(I1,J1)=-ANS(I,J)	EST 930
	118 CONTINUE	EST 940
C		EST 950
C	PRINT MATRICES A, B, C, D	EST 960
C		EST 970
	IF (DEBUG.NE.1) GO TO 123	EST 980
	WRITE (6,130)	EST 990
		EST 1000
		EST 1010
		EST 1020
		EST 1030

DO 120 I=1,NSU	EST 1040
120 WRITE (6,132) (A(I,J),J=1,NSU)	EST 1050
WRITE (6,134)	EST 1060
DO 122 I=1,NSU	EST 1070
122 WRITE (6,132) (B(I,J),J=1,NCS)	EST 1080
WRITE (6,136)	EST 1090
DO 124 I=1,NCS	EST 1100
124 WRITE (6,132) (C(I,J),J=1,NSU)	EST 1110
WRITE (6,133)	EST 1120
DO 126 I=1,NCS	EST 1130
126 WRITE (6,132) (D(I,J),J=1,NCS)	EST 1140
128 RETURN	EST 1150
C	EST 1160
130 FORMAT (1H1,5H MATRIX A)	EST 1170
132 FORMAT (X,11(E10.3,2X))	EST 1180
134 FORMAT (/5H MATRIX B)	EST 1190
135 FORMAT (/5H MATRIX C)	EST 1200
138 FORMAT (/5H MATRIX D)	EST 1210
C	EST 1220
END	EST 1230

SUBROUTINE FEALNC (A,N,IA,D,K,L)		FBC	10
C		FBC	20
C	*****	FBC	30
C	*	*FBC	40
C	***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	*FBC	50
C	* 1. BALANCE A REAL MATRIX A.	*FBC	60
C	*	*FBC	70
C	***** THIS SUB-PROGRAMS GLOSSARY OF FORTRAN NAMES:	*FBC	80
C	* A : MATRIX TO BE BALANCED	*FBC	90
C	* N : DIMENSION OF MATRIX A	*FBC	100
C	* IA : ROW DIMENSION OF A	*FBC	110
C	* D : ARRAY CONTAINING INFORMATION ABOUT PERMUTATIO	*FBC	120
C	* AND SCALE FACTORS	*FBC	130
C	* K,L : INTEGERS SUCH THAT A(I,J)=0 IF (1) I GT J AND	*FBC	140
C	* (2) J=1,2,...,K-1 OR I=L+1,...,N	*FBC	150
C	*	*FBC	160
C	*****	*FBC	170
C		FBC	180
	DIMENSION A(IA,1), D(1)	FBC	190
	DATA B/16.0/,D2/256.0/	FBC	200
	DATA ZERO/0.0/,ONE/1.0/,P95/.95/	FBC	210
C		FBC	220
C	***** REDUCE NORM A BY DIAGONAL SIMILARITY	FBC	230
C	***** TRANSFORMATION STORED IN D	FBC	240
C		FBC	250
	L1=1	FBC	260
	K1=N	FBC	270
C		FBC	280
C	***** SEARCH FOR ROWS ISOLATING AN EIGEN-	FBC	290
C	***** VALUE AND PUSH THEM DOWN	FBC	300
C		FBC	310
	101 K1P1=K1+1	FBC	320
	IF (K1.LT.1) GO TO 107	FBC	330
	K11=K1	FBC	340
	DO 106 JJ=1,K11	FBC	350
	J=K1P1-JJ	FBC	360
	R=ZERO	FBC	370
	DO 102 I=1,K1	FBC	380
	IF (I.EQ.J) GO TO 102	FBC	390
	R=R+ABS(A(J,I))	FBC	400
102	CONTINUE	FBC	410
	IF (R.NE.ZERO) GO TO 106	FBC	420
	D(K1)=J	FBC	430
	IF (J.EQ.K1) GO TO 105	FBC	440
	DO 103 I=1,K1	FBC	450
	F=A(I,J)	FBC	460
	A(I,J)=A(I,K1)	FBC	470
	A(I,K1)=F	FBC	480
103	CONTINUE	FBC	490
	DO 104 I=L1,N	FBC	500
	F=A(J,I)	FBC	510
	A(J,I)=A(K1,I)	FBC	520
	A(K1,I)=F	FBC	530
104	CONTINUE	FBC	540
105	K1=K1-1	FBC	550
	GO TO 101	FBC	560
	106 CONTINUE	FBC	570
C		FBC	580
C	***** SEARCH FOR COLUMNS ISOLATING AN	FBC	590
C	***** EIGENVALUE AND PUSH THEM LEFT	FBC	600

```

C
107 IF (K1.LT.L1) GO TO 113
    LL=L1
    DO 112 J=LL,K1
        C=ZERO
        DO 108 I=L1,K1
            IF (I.EQ.J) GO TO 108
            C=C+ABS(A(I,J))
108    CONTINUE
        IF (C.NE.ZERO) GO TO 112
        D(L1)=J
        IF (J.EQ.L1) GO TO 111
        DO 109 I=1,K1
            F=A(I,J)
            A(I,J)=A(I,L1)
            A(I,L1)=F
109    CONTINUE
        DO 110 I=L1,N
            F=A(J,I)
            A(J,I)=A(L1,I)
            A(L1,I)=F
110    CONTINUE
111    L1=L1+1
        GO TO 107
112 CONTINUE

```

```

C
C*****
C*****
C

```

NOW BALANCE THE SUBMATRIX IN ROWS
L1 THROUGH K1

```

113 K=L1
    L=K1
    IF (K1.LT.L1) GO TO 115
    DO 114 I=L1,K1
        D(I)=ONE
114    CONTINUE
115 NOCONV=0
    IF (K1.LT.L1) GO TO 124
    DO 123 I=L1,K1
        C=ZERO
        R=ZERO
        DO 116 J=L1,K1
            IF (J.EQ.I) GO TO 116
            C=C+ABS(A(J,I))
            R=R+ABS(A(I,J))
116    CONTINUE
        G=R/B
        F=ONE
        S=C+R
117    IF (C.GE.G) GO TO 118
        F=F*B
        C=C*B2
        GO TO 117
118    G=R*B
119    IF (C.LT.G) GO TO 120
        F=F/B
        C=C/B2
        GO TO 119

```

```

C
C*****
C

```

NOW BALANCE

```

FDC 610
FDC 620
FDC 630
FDC 640
FDC 650
FDC 660
FDC 670
FDC 680
FDC 690
FDC 700
FDC 710
FDC 720
FDC 730
FDC 740
FDC 750
FDC 760
FDC 770
FDC 780
FDC 790
FDC 800
FDC 810
FDC 820
FDC 830
FDC 840
FDC 850
FDC 860
FDC 870
FDC 880
FDC 890
FDC 900
FDC 910
FDC 920
FDC 930
FDC 940
FDC 950
FDC 960
FDC 970
FDC 980
FDC 990
FDC 1000
FDC 1010
FDC 1020
FDC 1030
FDC 1040
FDC 1050
FDC 1060
FDC 1070
FDC 1080
FDC 1090
FDC 1100
FDC 1110
FDC 1120
FDC 1130
FDC 1140
FDC 1150
FDC 1160
FDC 1170
FDC 1180
FDC 1190
FDC 1200

```

120	IF ((C+R)/F.GE.P95*S) GO TO 123	FBC 1210
	G=ONE/F	FBC 1220
	D(I)=D(I)*F	FBC 1230
	NOCONV=1	FBC 1240
	DO 121 J=L1,N	FBC 1250
	A(I,J)=A(I,J)*G	FBC 1260
121	CONTINUE	FBC 1270
	DO 122 J=1,K1	FBC 1280
	A(J,I)=A(J,I)*F	FBC 1290
122	CONTINUE	FBC 1300
123	CONTINUE	FBC 1310
124	IF (NOCONV.EQ.1) GO TO 115	FBC 1320
	RETURN	FBC 1330
C		FBC 1340
	END	FBC 1350
	SUBROUTINE FEVEU (A,N,IA,W,Z,WK,IER)	FEU 10
C		FEU 20
C	*****	FEU 30
C	*	*FEU 40
C	***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	*FEU 50
C	* 1. ACT AS THE EXECUTIVE CALLING PROGRAM FOR OBTAINING	*FEU 60
C	* THE EIGENVALUES-EIGENVECTORS OF A REAL MATRIX.	*FEU 70
C	*	*FEU 80
C	***** THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINES:	*FEU 90
C	* 1. FBALNC	*FEU 100
C	* 2. FRDHSS	*FEU 110
C	* 3. FBKXM1	*FEU 120
C	* 4. FBKXM2	*FEU 130
C	* 5. FORALG	*FEU 140
C	* 6. FERTST	*FEU 150
C	*	*FEU 160
C	***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	*FEU 170
C	* A : MATRIX WHOSE EIGENVALUES-EIGENVECTORS ARE TO	*FEU 180
C	* TO BE FOUND	*FEU 190
C	* N : DIMENSION OF MATRIX A	*FEU 200
C	* IA : ROW DIMENSION OF A	*FEU 210
C	* W : ARRAY CONTAINING THE EIGENVALUES	*FEU 220
C	* Z : MODAL MATRIX	*FEU 230
C	* WK : WORK ARRAY	*FEU 240
C	* IER : ERROR PARAMETER	*FEU 250
C	*	*FEU 260
C	*****	FEU 270
C		FEU 280
	DIMENSION A(IA,1), W(1), WK(N,1), Z(1)	FEU 290
	DATA ZERO,ONE/0.0,1.0/	FEU 300
C		FEU 310
C	***** INITIALIZE ERROR PARAMETERS	FEU 320
C		FEU 330
	IER=0	FEU 340
	JER=0	FEU 350
	IZ=IA	FEU 360
	IZZ=IZ+IZ	FEU 370
C		FEU 380
C	***** PACK A INTO AN N BY N ARRAY	FEU 390
C		FEU 400
	K=1	FEU 410
	L=1	FEU 420
	DO 105 J=1,N	FEU 430
	DO 105 I=1,N	FEU 440
	A(K,L)=A(I,J)	FEU 450

K=K+1	FEU 450
IF (K.GT.1A) K=1	FEU 470
IF (K.EQ.1) L=L+1	FEU 480
105 CONTINUE	FEU 490
N1=1	FEU 500
N2=N1+1	FEU 510
C	FEU 520
C*****	FEU 530
C	FEU 540
CALL FBALNC (A,N,N,WK(1,N1),K,L)	FEU 550
C	FEU 560
C*****	FEU 570
C*****	FEU 580
C	FEU 590
CALL FRDHSS (A,K,L,N,N,WK(1,N2))	FEU 600
C	FEU 610
C*****	FEU 620
C	FEU 630
II=1	FEU 640
JJ=1	FEU 650
NP1=N+1	FEU 660
DO 115 I=1,N	FEU 670
DO 110 J=1,N	FEU 680
Z(II)=ZERO	FEU 690
II=II+1	FEU 700
110 CONTINUE	FEU 710
Z(JJ)=ONE	FEU 720
JJ=JJ+NP1	FEU 730
115 CONTINUE	FEU 740
CALL FBKXM1 (Z,A,WK(1,N2),N,N,K,L)	FEU 750
II2=N	FEU 760
CALL FORALG (A,N,N,K,L,W(1),W(N+1),Z,II2,JER)	FEU 770
IF (JER.GT.128) GO TO 120	FEU 780
CALL FBKXM2 (WK(1,N1),Z,K,L,N,N,N)	FEU 790
C	FEU 800
C*****	FEU 810
C*****	FEU 820
C	FEU 830
120 DO 125 I=1,N	FEU 840
NP1=N+1	FEU 850
WK(I,N1)=W(NP1)	FEU 860
125 CONTINUE	FEU 870
JW=N+N	FEU 880
J=N	FEU 890
DO 130 I=1,N	FEU 900
W(JW-1)=W(J)	FEU 910
W(JW)=WK(J,N1)	FEU 920
JW=JW-2	FEU 930
J=J-1	FEU 940
130 CONTINUE	FEU 950
C	FEU 960
C*****	FEU 970
C*****	FEU 980
C	FEU 990
J=N	FEU 1000
135 IF (J.LT.1) GO TO 160	FEU 1010
IF (W(J+J).EQ.ZERO) GO TO 150	FEU 1020
C	FEU 1030
C*****	FEU 1040
C*****	FEU 1050

C	IS=I22*(J-1)+1	FEU 1060
	IG=N*(J-2)+1	FEU 1070
	IGZ=IG+N	FEU 1080
C		FEU 1090
C*****	MOVE COMPLEX CONJUGATE EIGENVECTOR	FEU 1100
C		FEU 1110
	DO 140 I=1,N	FEU 1120
	Z(IS)=Z(IG)	FEU 1130
	Z(IS+1)=-Z(IGZ)	FEU 1140
	IS=IS+2	FEU 1150
	IG=IG+1	FEU 1160
	IGZ=IGZ+1	FEU 1170
	140 CONTINUE	FEU 1180
C		FEU 1190
C*****	MOVE COMPLEX EIGENVECTOR	FEU 1200
C		FEU 1210
	IS=I22*(J-2)+1	FEU 1220
	IG=IS+I22	FEU 1230
	DO 145 I=1,N	FEU 1240
	Z(IS)=Z(IG)	FEU 1250
	Z(IS+1)=-Z(IG+1)	FEU 1260
	IS=IS+2	FEU 1270
	IG=IG+2	FEU 1280
	145 CONTINUE	FEU 1290
	J=J-2	FEU 1300
	GO TO 135	FEU 1310
C		FEU 1320
C*****	MOVE REAL EIGENVECTOR	FEU 1330
C		FEU 1340
	150 IS=I22*(J-1)+N+N	FEU 1350
	IG=N*J	FEU 1360
	DO 155 I=1,N	FEU 1370
	Z(IS-1)=Z(IG)	FEU 1380
	Z(IS)=ZERO	FEU 1390
	IS=IS-2	FEU 1400
	IG=IG-1	FEU 1410
	155 CONTINUE	FEU 1420
	J=J-1	FEU 1430
	GO TO 135	FEU 1440
C		FEU 1450
C*****	WRITE ERROR MESSAGES, IF ANY	FEU 1460
C		FEU 1470
	160 IF (IER.NE.0) CALL FERTST (IER,6HFEU)	FEU 1480
	IF (JER.EQ.0) GO TO 165	FEU 1490
	IER=JER	FEU 1500
	CALL FERTST (IER,6HFEU)	FEU 1510
	165 RETURN	FEU 1520
C		FEU 1530
	END	FEU 1540
	SUBROUTINE FBKXM1 (Z,H,D,MM,IZH,K,L)	FEU 1550
C		FM1 10
C*****		FM1 20
C		FM1 30
C		FM1 40
C*****	THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	FM1 50
C	1. BACKTRANSFORM THE EIGENVECTORS OF THE UPPER HESSENBERG	FM1 60
C	MATRIX.	FM1 70
C		FM1 80
C*****	THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	FM1 90
C	Z : EIGENVECTORS OF MATRIX A	FM1 100

C	*	H	: SUB-DIAGONAL ELEMENTS USED FOR STORING BACK-	*FM1	110	
C	*		TRANSFORMATION INFORMATION	*FM1	120	
C	*	D	: DETAILS OF THE TRANSFORMATION	*FM1	130	
C	*	MM	: NUMBER OF COLUMNS IN MATRIX Z	*FM1	140	
C	*	IZH	: ROW DIMENSION OF MATRICES Z AND H	*FM1	150	
C	*	K,L	: SAME AS IN SUBROUTINE FBKXM1	*FM1	160	
C	*			*FM1	170	
C	*****				*FM1	180
C				FM1	190	
		DIMENSION Z(IZH,1), H(IZH,1), D(1)		FM1	200	
		DATA ZERO,ONE/0.0,1.0/		FM1	210	
		LM2=L-2		FM1	220	
		IF (LM2.LT.K) GO TO 107		FM1	230	
		LTEMP=LM2+K		FM1	240	
		DO 106 KI=K,LM2		FM1	250	
		M=LTEMP-KI		FM1	260	
		MA=M+1		FM1	270	
		T=H(MA,M)		FM1	280	
		IF (T.EQ.ZERO) GO TO 106		FM1	290	
		T=T*D(MA)		FM1	300	
		MP2=M+2		FM1	310	
		IF (MP2.GT.L) GO TO 102		FM1	320	
		DO 101 I=MP2,L		FM1	330	
		D(I)=H(I,M)		FM1	340	
101		CONTINUE		FM1	350	
102		IF (MA.GT.L) GO TO 106		FM1	360	
		TINV=ONE/T		FM1	370	
		DO 105 J=1,MM		FM1	380	
		G=ZERO		FM1	390	
		DO 103 I=MA,L		FM1	400	
		G=G+D(I)*Z(I,J)		FM1	410	
103		CONTINUE		FM1	420	
		G=G*TINV		FM1	430	
		DO 104 I=MA,L		FM1	440	
		Z(I,J)=Z(I,J)+G*D(I)		FM1	450	
104		CONTINUE		FM1	460	
105		CONTINUE		FM1	470	
106		CONTINUE		FM1	480	
107		RETURN		FM1	490	
C				FM1	500	
		END		FM1	510	
		SUBROUTINE FBKXM2 (D,Z,K,L,MM,N,IZ)		FM2	10	
C				FM2	20	
C	*****				*FM2	30
C	*			*FM2	40	
C	***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:				*FM2	50
C	*	1. BACKTRANSFORM THE EIGENVECTORS OF A BALANCED MATRIX		*FM2	60	
C	*			*FM2	70	
C	***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:				*FM2	80
C	*	D	: INFORMATION ON THE DETAILS OF TRANSFORMATION	*FM2	90	
C	*	Z	: AT ENTRANCE: MODAL MATRIX TO BE TRANSFORMED	*FM2	100	
C	*		AT EXIT, TRANSFORMED MODAL MATRIX	*FM2	110	
C	*	K	: ROW,COLUMN INDEX OF STARTING ELEMENT TO BE	*FM2	120	
C	*		TRANSFORMED	*FM2	130	
C	*	L	: ROW,COLUMN INDEX OF LAST ELEMENT TO BE TRANS-	*FM2	140	
C	*		FORMED	*FM2	150	
C	*	MM	: NUMBER OF COLUMNS IN MATRIX Z	*FM2	160	
C	*	N	: NUMBER OF ROWS IN Z = LENGTH OF VECTOR D	*FM2	170	
C	*	IZ	: ROW DIMENSION OF Z	*FM2	180	
C	*			*FM2	190	

C*****	FM2	200
C	FM2	210
DIMENSION Z(IZ,1), D(1)	FM2	220
C	FM2	230
C*****	FM2	240
C*****	FM2	250
C	FM2	260
DO 101 I=K,L	FM2	270
S=D(I)	FM2	280
DO 101 J=1,MM	FM2	290
Z(I,J)=Z(I,J)*S	FM2	300
101 CONTINUE	FM2	310
C	FM2	320
C*****	FM2	330
C*****	FM2	340
C	FM2	350
IF (K.EQ.1) GO TO 104	FM2	360
KM1=K-1	FM2	370
DO 103 I=1,KM1	FM2	380
II=K-I	FM2	390
JJ=D(II)	FM2	400
IF (II.EQ.JJ) GO TO 103	FM2	410
DO 102 J=1,MM	FM2	420
S=Z(II,J)	FM2	430
Z(II,J)=Z(JJ,J)	FM2	440
Z(JJ,J)=S	FM2	450
102 CONTINUE	FM2	460
103 CONTINUE	FM2	470
104 IF (L.EQ.N) GO TO 107	FM2	480
LP1=L+1	FM2	490
DO 105 II=LP1,N	FM2	500
JJ=D(II)	FM2	510
IF (II.EQ.JJ) GO TO 105	FM2	520
DO 105 J=1,MM	FM2	530
S=Z(II,J)	FM2	540
Z(II,J)=Z(JJ,J)	FM2	550
Z(JJ,J)=S	FM2	560
105 CONTINUE	FM2	570
106 CONTINUE	FM2	580
107 RETURN	FM2	590
C	FM2	600
END	FM2	610
SUBROUTINE FERTST (IER,NAME)	FER	10
C	FER	20
C*****	FER	30
C	*FER	40
C	*FER	50
C	*FER	60
C*****	*FER	70
C	*FER	80
C	*FER	90
C	*FER	100
C*****	FER	110
C	FER	120
DIMENSION ITP(2,4), IBIT(4)	FER	130
INTEGER WARN,WARF,TERM,PRINTR	FER	140
EQUIVALENCE (IBIT(1),WARN), (IBIT(2),WARF), (IBIT(3),TERM)	FER	150
DATA ITP/10HWARNING,10H,10HWARNING(WI,10HTH FIX),FER	160	
110:TERMINAL,10H,10HNON-DEFINE,10HD,IBIT/32,6FER	170	
24,128,0/	FER	180

```

IERR=IER
IF (IERR.GE.WARN) GO TO 101
C*****
C                                     NON-DEFINED
C
C      IERK=4
C      GO TO 104
C 101 IF (IERR.LT.TERM) GO TO 102
C*****
C                                     TERMINAL
C
C      IERK=3
C      GO TO 104
C 102 IF (IERR.LT.WARF) GO TO 103
C*****
C                                     WARNING(WITH FIX)
C
C      IERK=2
C      GO TO 104
C*****
C                                     WARNING
C
C 103 IERK=1
C*****
C                                     EXTRACT *N*
C
C 104 IERR=IERR-IBIT(IERK)
C*****
C                                     PRINT ERROR MESSAGE
C
C      WRITE (6,105) (ITYP(I,IERK),I=1,2),NAME,IERR,IER
C      RETURN
C
C 105 FORMAT (1H0,2A10,4X,A6,4X,I2,8H (IER = ,I3,1H))
C
C      END
C      SUBROUTINE FORALG (HS,N,IH,K,L,WRL,WIM,Z,IZ,IER)
C*****
C      *
C      * THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:
C      * 1. FIND THE EIGENVALUES AND EIGENVECTORS OF THE UPPER
C      * HESSENBERG MATRIX.
C      *
C      * THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINE:
C      * 1. FERTST
C      *
C      * THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:
C      * ALL VARIABLE NAMES AND ARRAYS ARE AS DEFINED IN SUBROUTINES
C      * FEUEV,FBALNC,FBKXM1,FBKXM2.
C*****
C
C      DIMENSION HS(IH,N), WRL(N), WIM(N), Z(IZ,N), T3(2)
C      LOGICAL NTL5
C      COMPLEX Z3
C      EQUIVALENCE (Z3,T3(1))
C      DATA RDEL/16414000000000000000B/
C      DATA P4/0.4375/,P5/0.5/,P7/0.75/,ZERO/0.0/,ONE/1.0/
C      IER=0

```

C		FOR 250
C*****	STORE ROOTS ISOLATED BY FBALNC	FOR 260
C		FOR 270
	DO 101 I=1,N	FOR 280
	IF (I.GE.K.AND.I.LE.L) GO TO 101	FOR 290
	WRL(I)=HS(I,I)	FOR 300
	WIM(I)=ZERO	FOR 310
101	CONTINUE	FOR 320
	IEN=L	FOR 330
	T=ZERO	FOR 340
C		FOR 350
C*****	SEARCH FOR NEXT EIGENVALUES	FOR 360
C		FOR 370
	102 IF (IEN.LT.K) GO TO 123	FOR 380
	ITS=0	FOR 390
	NA=IEN-1	FOR 400
	IENM2=NA-1	FOR 410
C		FOR 420
C*****	LOOK FOR SINGLE SMALL SUB-DIAGONAL	FOR 430
C*****	ELEMENT	FOR 440
C		FOR 450
	103 NPL=IEN+K	FOR 460
	DO 104 LL=K,IEN	FOR 470
	LB=NPL-LL	FOR 480
	IF (LB.EQ.K) GO TO 105	FOR 490
	IF (ABS(HS(LB,LB-1)).LE.RDELP*(ABS(HS(LB-1,LB-1))+ABS(HS(LB,LB)))) GO TO 105	FOR 500
1))) GO TO 105	FOR 510
104	CONTINUE	FOR 520
C		FOR 530
C*****		FOR 540
C		FOR 550
	105 X=HS(IEN,IEN)	FOR 560
	IF (LB.EQ.IEN) GO TO 121	FOR 570
	Y=HS(NA,NA)	FOR 580
	W=HS(IEN,NA)*HS(NA,IEN)	FOR 590
	IF (LB.EQ.NA) GO TO 122	FOR 600
	IF (ITS.EQ.30) GO TO 151	FOR 610
C		FOR 620
C*****	FORM SHIFT	FOR 630
C		FOR 640
	IF (ITS.NE.10.AND.ITS.NE.20) GO TO 107	FOR 650
	T=T+X	FOR 660
	DO 106 I=K,IEN	FOR 670
	HS(I,I)=HS(I,I)-X	FOR 680
106	CONTINUE	FOR 690
	S=ABS(HS(IEN,NA))+ABS(HS(NA,IENM2))	FOR 700
	X=P7*S	FOR 710
	Y=X	FOR 720
	W=-P4*S*S	FOR 730
107	ITS=ITS+1	FOR 740
C		FOR 750
C*****	LOOK FOR TWO CONSECUTIVE SMALL	FOR 760
C*****	SUB-DIAGONAL ELEMENTS	FOR 770
C		FOR 780
	NAML=IENM2+LB	FOR 790
	DO 108 MM=LB,IENM2	FOR 800
	M=NAML-MM	FOR 810
	ZZ=HS(M,M)	FOR 820
	R=X-ZZ	FOR 830
	S=Y-ZZ	FOR 840

P=(R*S-W)/HS(M+1,M)+HS(M,M+1)	FCR 050
Q=HS(M+1,M+1)-ZZ-R-S	FCR 060
R=HS(M+2,M+1)	FCR 070
S=ABS(P)+ABS(Q)+ABS(R)	FCR 080
P=P/S	FCR 090
Q=Q/S	FCR 100
R=R/S	FCR 110
IF (M.EQ.LD) GO TO 109	FCR 120
IF (ABS(HS(M,M-1))*(ABS(Q)+ABS(R)).LE.RDELP*ABS(P)*(ABS(HS(M-1,M-1))+ABS(ZZ)+ABS(HS(M+1,M+1)))) GO TO 109	FCR 130
108 CONTINUE	FCR 140
109 MP2=M+2	FCR 150
DO 110 I=MP2,IEN	FCR 160
HS(I,I-2)=ZERO	FCR 170
IF (I.EQ.MP2) GO TO 110	FCR 180
HS(I,I-3)=ZERO	FCR 190
110 CONTINUE	FCR 200
C	FCR 210
C*****	FCR 220
C*****	FCR 230
C	FCR 240
DO 120 KA=M,NA	FCR 250
NTLS=KA,NE,NA	FCR 260
IF (KA.EQ.M) GO TO 111	FCR 270
P=HS(KA,KA-1)	FCR 280
Q=HS(KA+1,KA-1)	FCR 290
R=ZERO	FCR 300
IF (NTLS) R=HS(KA+2,KA-1)	FCR 310
X=ABS(P)+ABS(Q)+ABS(R)	FCR 320
IF (X.EQ.ZERO) GO TO 120	FCR 330
P=P/X	FCR 340
Q=Q/X	FCR 350
R=R/X	FCR 360
111 CONTINUE	FCR 370
S=SIGN(SORT(P*P+Q*Q+R*R),P)	FCR 380
IF (KA.EQ.M) GO TO 112	FCR 390
HS(KA,KA-1)=-S*X	FCR 400
GO TO 113	FCR 410
112 IF (LD.NE.M) HS(KA,KA-1)=-HS(KA,KA-1)	FCR 420
113 P=P+S	FCR 430
X=P/S	FCR 440
Y=Q/S	FCR 450
ZZ=R/S	FCR 460
Q=Q/P	FCR 470
R=R/P	FCR 480
C	FCR 490
C*****	FCR 500
C	FCR 510
DO 115 J=KA,N	FCR 520
P=HS(KA,J)+Q*HS(KA+1,J)	FCR 530
IF (.NOT.NTLS) GO TO 114	FCR 540
P=P+R*HS(KA+2,J)	FCR 550
HS(KA+2,J)=HS(KA+2,J)-P*ZZ	FCR 560
114 HS(KA+1,J)=HS(KA+1,J)-P*Y	FCR 570
HS(KA,J)=HS(KA,J)-P*X	FCR 580
115 CONTINUE	FCR 590
J=MIN0(IEN,KA+3)	FCR 600
C	FCR 610
C*****	FCR 620
C	FCR 630
	FCR 640

DO 117 I=1,J	FOR 1450
P=X*HS(I,KA)+Y*HS(I,KA+1)	FOR 1460
IF (.NOT.NTLS) GO TO 116	FOR 1470
P=P+ZZ*HS(I,KA+2)	FOR 1480
HS(I,KA+2)=HS(I,KA+2)-P*R	FOR 1490
116 HS(I,KA+1)=HS(I,KA+1)-P*Q	FOR 1500
HS(I,KA)=HS(I,KA)-P	FOR 1510
117 CONTINUE	FOR 1520
IF (IZ.LT.N) GO TO 120	FOR 1530
C	FOR 1540
C***** ACCUMULATE TRANSFORMATIONS	FOR 1550
C	FOR 1560
DO 119 I=K,L	FOR 1570
P=X*Z(I,KA)+Y*Z(I,KA+1)	FOR 1580
IF (.NOT.NTLS) GO TO 118	FOR 1590
P=P+ZZ*Z(I,KA+2)	FOR 1600
Z(I,KA+2)=Z(I,KA+2)-P*R	FOR 1610
118 Z(I,KA+1)=Z(I,KA+1)-P*Q	FOR 1620
Z(I,KA)=Z(I,KA)-P	FOR 1630
119 CONTINUE	FOR 1640
120 CONTINUE	FOR 1650
GO TO 103	FOR 1660
C	FOR 1670
C***** ONE ROOT FOUND	FOR 1680
C	FOR 1690
121 HS(IEN,IEN)=X+T	FOR 1700
WRL(IEN)=HS(IEN,IEN)	FOR 1710
WIM(IEN)=ZERO	FOR 1720
IEN=NA	FOR 1730
GO TO 102	FOR 1740
C	FOR 1750
C***** TWO ROOTS FOUND	FOR 1760
C	FOR 1770
122 P=(Y-X)*P5	FOR 1780
Q=P*P*W	FOR 1790
ZZ=SQRT(ABS(Q))	FOR 1800
HS(IEN,IEN)=X+T	FOR 1810
X=HS(IEN,IEN)	FOR 1820
HS(NA,NA)=Y+T	FOR 1830
IF (Q.LT.ZERO) GO TO 126	FOR 1840
C	FOR 1850
C***** REAL PAIR	FOR 1860
C	FOR 1870
ZZ=P+SIGN(ZZ,P)	FOR 1880
WRL(NA)=X+ZZ	FOR 1890
WRL(IEN)=WRL(NA)	FOR 1900
IF (ZZ.NE.ZERO) WRL(IEN)=X-W/ZZ	FOR 1910
WIM(NA)=ZERO	FOR 1920
WIM(IEN)=ZERO	FOR 1930
X=HS(IEN,NA)	FOR 1940
R=SQRT(X*X+ZZ*ZZ)	FOR 1950
P=X/R	FOR 1960
Q=ZZ/R	FOR 1970
C	FOR 1980
C***** ROW MODIFICATION	FOR 1990
C	FOR 2000
DO 123 J=NA,N	FOR 2010
ZZ=HS(NA,J)	FOR 2020
HS(NA,J)=Q*ZZ+P*HS(IEN,J)	FOR 2030
HS(IEN,J)=Q*HS(IEN,J)-P*ZZ	FOR 2040

123 CONTINUE		
C		FOR 2050
C*****	COLUMN MODIFICATION	FOR 2060
C		FOR 2070
DO 124 I=1,IEN		FOR 2080
ZZ=HS(I,NA)		FOR 2090
HS(I,NA)=Q*ZZ+P*HS(I,IEN)		FOR 2100
HS(I,IEN)=Q*HS(I,IEN)-P*ZZ		FOR 2110
124 CONTINUE		FOR 2120
IF (I2.LT.N) GO TO 127		FOR 2130
C		FOR 2140
C*****	ACCUMULATE TRANSFORMATIONS	FOR 2150
C		FOR 2160
DO 125 I=K,L		FOR 2170
ZZ=Z(I,NA)		FOR 2180
Z(I,NA)=Q*ZZ+P*Z(I,IEN)		FOR 2190
Z(I,IEN)=Q*Z(I,IEN)-P*ZZ		FOR 2200
125 CONTINUE		FOR 2210
GO TO 127		FOR 2220
C		FOR 2230
C*****	COMPLEX PAIR	FOR 2240
C		FOR 2250
126 WRL(NA)=X+P		FOR 2260
WRL(IEN)=X+P		FOR 2270
WIM(NA)=ZZ		FOR 2280
WIM(IEN)=-ZZ		FOR 2290
127 IEN=IENM2		FOR 2300
GO TO 102		FOR 2310
C		FOR 2320
C*****	ALL ROOTS FOUND, NOW	FOR 2330
C*****	BACKSUBSTITUTE	FOR 2340
C		FOR 2350
128 IF (I2.LT.N) GO TO 156		FOR 2360
RNORM=ZERO		FOR 2370
KA=1		FOR 2380
DO 130 I=1,N		FOR 2390
DO 129 J=KA,N		FOR 2400
RNORM=RNORM+ABS(HS(I,J))		FOR 2410
129 CONTINUE		FOR 2420
KA=I		FOR 2430
130 CONTINUE		FOR 2440
IF (RNORM.EQ.ZERO) GO TO 156		FOR 2450
DO 145 NN=1,N		FOR 2460
IEN=N+1-NN		FOR 2470
P=WRL(IEN)		FOR 2480
Q=WIM(IEN)		FOR 2490
NA=IEN-1		FOR 2500
IF (Q.GT.ZERO) GO TO 145		FOR 2510
IF (Q.LT.ZERO) GO TO 137		FOR 2520
C		FOR 2530
C*****	REAL VECTOR	FOR 2540
C		FOR 2550
M=IEN		FOR 2560
HS(IEN,IEN)=ONE		FOR 2570
IF (NA.EQ.0) GO TO 145		FOR 2580
DO 136 II=1,NA		FOR 2590
I=IEN-II		FOR 2600
W=HS(I,I)-P		FOR 2610
R=HS(I,IEN)		FOR 2620
IF (R.GT.NA) GO TO 132		FOR 2630
		FOR 2640

	DO 131 J=M,NA	FOR 2650
	R=R+HS(I,J)*HS(J,IEN)	FOR 2660
131	CONTINUE	FOR 2670
132	IF (WIM(I).GE.ZERO) GO TO 133	FOR 2680
	ZZ=W	FOR 2690
	S=R	FOR 2700
	GO TO 136	FOR 2710
133	M=I	FOR 2720
	IF (WIM(I).NE.ZERO) GO TO 134	FOR 2730
	T=W	FOR 2740
	IF (W.EQ.ZERO) T=RDELP*RNORM	FOR 2750
	HS(I,IEN)=-R/T	FOR 2760
	GO TO 136	FOR 2770
C		FOR 2780
C*****	SOLVE REAL EQUATIONS	FOR 2790
C		FOR 2800
134	X=HS(I,I+1)	FOR 2810
	Y=HS(I+1,I)	FOR 2820
	Q=(WRL(I)-P)*(WRL(I)-P)+WIM(I)*WIM(I)	FOR 2830
	T=(X*S-ZZ*R)/Q	FOR 2840
	HS(I,IEN)=T	FOR 2850
	IF (ABS(X).LE.ABS(ZZ)) GO TO 135	FOR 2860
	HS(I+1,IEN)=(-R-W*T)/X	FOR 2870
	GO TO 136	FOR 2880
135	HS(I+1,IEN)=(-S-Y*T)/ZZ	FOR 2890
136	CONTINUE	FOR 2900
C		FOR 2910
C*****	END REAL VECTOR	FOR 2920
C		FOR 2930
	GO TO 145	FOR 2940
C		FOR 2950
C*****	LAST VECTOR COMPONENT CHOSEN	FOR 2960
C*****	IMAGINARY SO THAT EIGENVECTOR	FOR 2970
C*****	MATRIX IS TRIANGULAR	FOR 2980
C		FOR 2990
137	M=NA	FOR 3000
C		FOR 3010
C*****	COMPLEX VECTOR	FOR 3020
C		FOR 3030
	IF (ABS(HS(IEN,NA)).LE.ABS(HS(NA,IEN))) GO TO 138	FOR 3040
	HS(NA,NA)=Q/HS(IEN,NA)	FOR 3050
	HS(NA,IEN)=-HS(IEN,IEN)-P/HS(IEN,NA)	FOR 3060
	GO TO 139	FOR 3070
138	CONTINUE	FOR 3080
	Z3=CMPLX(ZERO,-HS(NA,IEN))/CMPLX(HS(NA,NA)-P,Q)	FOR 3090
	HS(NA,NA)=T3(1)	FOR 3100
	HS(NA,IEN)=T3(2)	FOR 3110
139	HS(IEN,NA)=ZERO	FOR 3120
	HS(IEN,IEN)=ONE	FOR 3130
	IENM2=NA-1	FOR 3140
	IF (IENM2.EQ.0) GO TO 145	FOR 3150
	DO 144 II=1,IENM2	FOR 3160
	I=NA-II	FOR 3170
	W=HS(I,I)-P	FOR 3180
	RA=ZERO	FOR 3190
	SA=HS(I,IEN)	FOR 3200
	DO 140 J=I,NA	FOR 3210
	RA=RA+HS(I,J)*HS(J,NA)	FOR 3220
	SA=SA+HS(I,J)*HS(J,IEN)	FOR 3230
140	CONTINUE	FOR 3240

	IF (NIM(I).GE.ZERO) GO TO 141	FOR 3350
	ZZ=N	FOR 3360
	R=RA	FOR 3370
	S=SA	FOR 3380
	GO TO 144	FOR 3390
141	M=I	FOR 3400
	IF (NIM(I).NE.ZERO) GO TO 142	FOR 3410
	Z3=CMPLX(-RA,-SA)/CMPLX(I,0)	FOR 3420
	HS(I,NA)=TS(1)	FOR 3430
	HS(I,IEN)=TS(2)	FOR 3440
	GO TO 144	FOR 3450
C		FOR 3460
C*****	SOLVE COMPLEX EQUATIONS	FOR 3470
C		FOR 3480
142	X=HS(I,I+1)	FOR 3490
	Y=HS(I+1,I)	FOR 3500
	UR=(URL(I)-P)*(URL(I)-P)+NIM(I)*NIM(I)-Q*Q	FOR 3510
	UI=(URL(I)-P)*Q	FOR 3520
	UI=UI*UI	FOR 3530
	IF (UR.EQ.ZERO.AND.UI.EQ.ZERO) UR=RDELP*RNORM*(ABS(W)+ABS(O)	FOR 3540
1	+ABS(X)+ABS(Y)+ABS(ZZ))	FOR 3550
	Z3=CMPLX((X-R-ZZ*RA+Q*SA,Y-S-ZZ*SA-Q*RA)/CMPLX(UR,UI)	FOR 3560
	HS(I,NA)=TS(1)	FOR 3570
	HS(I,IEN)=TS(2)	FOR 3580
	IF (ABS(X).LE.ABS(ZZ)+ABS(O)) GO TO 143	FOR 3590
	HS(I+1,NA)=(-RA-U*HS(I,NA)+Q*HS(I,IEN))/X	FOR 3600
	HS(I+1,IEN)=(-SA-U*HS(I,IEN)+Q*HS(I,NA))/X	FOR 3610
	GO TO 143	FOR 3620
143	CONTINUE	FOR 3630
	Z3=CMPLX(-R-V*HS(I,NA),-S-V*HS(I,IEN))/CMPLX(ZZ,0)	FOR 3640
	HS(I+1,NA)=TS(1)	FOR 3650
	HS(I+1,IEN)=TS(2)	FOR 3660
144	CONTINUE	FOR 3670
C		FOR 3680
C*****	END COMPLEX VECTOR	FOR 3690
C		FOR 3700
145	CONTINUE	FOR 3710
C		FOR 3720
C*****	END BACKSUBSTITUTION	FOR 3730
C*****	VECTORS OF ISOLATED ROOTS	FOR 3740
C		FOR 3750
	DO 147 I=1,N	FOR 3760
	IF (I.GE.K.AND.I.LE.L) GO TO 147	FOR 3770
	DO 146 J=1,N	FOR 3780
	Z(I,J)=HS(I,J)	FOR 3790
146	CONTINUE	FOR 3800
147	CONTINUE	FOR 3810
	IF (L.EQ.0) GO TO 155	FOR 3820
C		FOR 3830
C*****	MULTIPLY BY TRANSFORMATION MATRIX	FOR 3840
C		FOR 3850
	DO 150 JJ=K,N	FOR 3860
	J=J+K-JJ	FOR 3870
	H=HEND(J,L)	FOR 3880
	DO 148 I=K,L	FOR 3890
	ZZ=ZERO	FOR 3900
	DO 149 KA=K,M	FOR 3910
	Z3=Z3+Z(I,KA)*HS(KA,J)	FOR 3920
148	CONTINUE	FOR 3930
	Z(I,J)=ZZ	FOR 3940

149	CONTINUE	FOR 3850
150	CONTINUE	FOR 3860
	GO TO 153	FOR 3870
C		FOR 3880
C*****	NO CONVERGENCE AFTER 30 ITERATIONS	FOR 3890
C*****	SET ERROR INDICATOR TO THE INDEX	FOR 3900
C*****	OF THE CURRENT EIGENVALUE	FOR 3910
C		FOR 3920
151	ITER=150+KEN	FOR 3930
	DO 152 I=1, IEM	FOR 3940
	WRL(I)=ZERO	FOR 3950
	WEM(I)=ZERO	FOR 3960
152	CONTINUE	FOR 3970
	IF (1/2*(L7+H)) GO TO 155	FOR 3980
	DO 153 J=1, H	FOR 3990
	DO 153 J=1, H	FOR 4000
	EXTJ=ZERO	FOR 4010
153	CONTINUE	FOR 4020
154	CONTINUE	FOR 4030
155	CONTINUE	FOR 4040
	CALL FERTST (ITER, SHFCRALS)	FOR 4050
156	RETURN	FOR 4060
C		FOR 4070
	END	FOR 4080
	SUBROUTINE FERMSS (A,K,L,M,IA,D)	FMS 10
C		FMS 20
C*****		FMS 30
C		FMS 40
C*****	THIS SUBROUTINE PERFORMS THE FOLLOWING FUNCTION:	FMS 50
C	1. REDUCE A REAL MATRIX TO UPPER HESSEBERG FORM THRU	FMS 60
C	ORTHOGONAL TRANSFORMATIONS.	FMS 70
C		FMS 80
C*****	THIS SUB PROGRAM'S LIBRARY OF FORTRAN NAMES:	FMS 90
C	A	FMS 100
C	1. MATRIX TO BE REDUCED TO HESSEBERG FORM	FMS 110
C	K	FMS 120
C	1. ORDER OF A	FMS 130
C	L	FMS 140
C	1. ROW DIMENSION OF MATRIX A	FMS 150
C	M	FMS 160
C	1. DETAILS OF TRANSFORMATION FOR SUBSEQUENT USE	FMS 170
C		FMS 180
	DIMENSION A(1:M,1:M), D(1:M)	FMS 190
	DATA ZERO	FMS 200
	IA=1	FMS 210
	IF (IA=1) GO TO 159	FMS 220
	DO 157 I=1, M	FMS 230
	DO 157 I=1, M	FMS 240
	DO 157 I=1, M	FMS 250
	DO 157 I=1, M	FMS 260
	DO 157 I=1, M	FMS 270
C		FMS 280
C*****	SCALE COLUMN	FMS 290
C		FMS 300
	DO 158 I=1, M	FMS 310
	DO 158 I=1, M+ABS(0.01*M-10)	FMS 320
151	CONTINUE	FMS 330
	IF (SCALE=0.0) GO TO 160	FMS 340
	DO 158 I=1, M	FMS 350
C		FMS 360
C*****	DO 10 I=L,M,-1	FMS 370

C	DO 102 II=M,L	FHS 370
	I=MP-II	FHS 380
	D(I)=A(I,M-1)/SCALE	FHS 390
	H=H+D(I)*D(I)	FHS 400
102	CONTINUE	FHS 410
	G=-SIGN(SQRT(H),D(M))	FHS 420
	H=H-D(M)*G	FHS 430
	D(M)=D(M)-G	FHS 440
	DO 105 J=M,N	FHS 450
	F=ZERO	FHS 460
C		FHS 470
C*****	DO 15 I=L,M,-1	FHS 480
C		FHS 490
	DO 103 II=M,L	FHS 500
	I=MP-II	FHS 510
	F=F+D(I)*A(I,J)	FHS 520
103	CONTINUE	FHS 530
	F=F/H	FHS 540
	DO 104 I=M,L	FHS 550
	A(I,J)=A(I,J)-F*D(I)	FHS 560
104	CONTINUE	FHS 570
105	CONTINUE	FHS 580
	DO 108 I=1,L	FHS 590
	F=ZERO	FHS 600
C		FHS 610
C*****	DO 30 J=L,M,-1	FHS 620
C		FHS 630
	DO 106 JJ=M,L	FHS 640
	J=MP-JJ	FHS 650
	F=F+D(J)*A(I,J)	FHS 660
106	CONTINUE	FHS 670
	F=F/H	FHS 680
	DO 107 J=M,L	FHS 690
	A(I,J)=A(I,J)-F*D(J)	FHS 700
107	CONTINUE	FHS 710
108	CONTINUE	FHS 720
	D(M)=SCALE*D(M)	FHS 730
	A(M,M-1)=SCALE*G	FHS 740
109	CONTINUE	FHS 750
110	RETURN	FHS 760
C		FHS 770
	END	FHS 780
		FHS 790

SUBROUTINE GZOC (N,EU,EVALS,BMAT,CMAT,DMAT,BH,CH,X,Y,NPORT,PR,MP,MGZC	10
1S)	GZC 20
C	GZC 30
C*****	GZC 40
C *	GZC 50
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTIONS:	GZC 60
C * 1. OBTAIN AND STORE COMPLETE INFORMATION ABOUT THE	GZC 70
C * OPEN-CIRCUIT IMPEDANCE MATRIX IN PARTIAL FRACTION	GZC 80
C * EXPANSION (PFE) FORM.	GZC 90
C * 2. CHECK OF JW-AXIS OR REPEATED EIGENVALUES.	GZC 100
C * 3. PRINT ENTRIES OF ZOC, IF REQUESTED.	GZC 110
C *	GZC 120
C***** THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINE:	GZC 130
C * 1. GZOCPR	GZC 140
C * 2. **** LINED4 **** LIBRARY DEPENDENT ROUTINE	GZC 150
C *	GZC 160
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	GZC 170
C * ALL VARIABLE NAMES AND ARRAYS AS DEFINED IN SUB-PROGRAM AMAIN	GZC 180
C *	GZC 190
C*****	GZC 200
C	GZC 210
COMPLEX EVALS(1),EU(MS,1),X(MS,1),Y(MS,1),SUM,BH(MS,1)	GZC 220
COMPLEX CMKP,1)	GZC 230
INTEGER ER,TYPE,PR	GZC 240
DIMENSION BMAT(MS,1), CMAT(MP,1), DMAT(MP,1), NPORT(1)	GZC 250
COMMON /ENOS/ NCAP,NDUS,NRES,NIND,NDCS,NCPRT	GZC 260
C	GZC 270
C*****OBTAIN THE INVERSE OF THE MODAL MATRIX	GZC 280
C	GZC 290
DO 104 I=1,N	GZC 300
DO 102 J=1,N	GZC 310
102 Y(I,J)=CMPLX(0.00,0.00)	GZC 320
104 Y(I,I)=CMPLX(1.00,0.00)	GZC 330
C	GZC 340
C*****SYSTEM DEPENDENT ROUTINE FOR FINDING INVERSE OF A COMPLEX	GZC 350
C***** MATRIX	GZC 360
C	GZC 370
CALL LINED4 (EU,Y,X,20,N,N,IERR)	GZC 380
IF (IERR.NE.0) GO TO 122	GZC 390
DO 105 I=1,N	GZC 400
DO 106 J=1,N	GZC 410
U=REAL(EU(I,J))	GZC 420
U=AIMAG(EU(I,J))	GZC 430
U1=REAL(X(I,J))	GZC 440
U1=AIMAG(X(I,J))	GZC 450
IF (ABS(U).LT.1.00E-15) U=0.0000	GZC 460
IF (ABS(U).LT.1.00E-15) U=0.0000	GZC 470
IF (ABS(U1).LT.1.00E-15) U1=0.000	GZC 480
IF (ABS(U1).LT.1.00E-15) U1=0.000	GZC 490
EU(I,J)=CMPLX(U,U)	GZC 500
X(I,J)=CMPLX(U1,U1)	GZC 510
105 CONTINUE	GZC 520
C	GZC 530
C***** FORM THE FINU*DMAT PRODUCT	GZC 540
C	GZC 550
DO 110 I=1,N	GZC 560
DO 110 J=1,NCPRT	GZC 570
SUM=CMPLX(0.0000,0.0000)	GZC 580
DO 103 K=1,N	GZC 590
103 SUM=SUM+X(I,K)*DMAT(K,J)	GZC 600

BH(I,J)=SUM	GZC	610
110 CONTINUE	GZC	620
C	GZC	630
C*****FORM THE PRODUCT C*T	GZC	640
C	GZC	650
DO 114 I=1,NCPRT	GZC	650
DO 114 J=1,N	GZC	670
SUM=CMPLX(0.00,0.00)	GZC	680
DO 112 K=1,N	GZC	690
112 SUM=SUM+CMAT(I,K)*EU(K,J)	GZC	700
CH(I,J)=SUM	GZC	710
114 CONTINUE	GZC	720
C	GZC	730
C***** CHECK FOR REPEATED OR JW-AXIS EIGENVALUES:	GZC	740
C	GZC	750
IWARN1=0	GZC	760
IWARN2=0	GZC	770
DO 120 I=1,N	GZC	780
U1=REAL(EVALS(I))	GZC	790
U1=AIMAG(EVALS(I))	GZC	800
J=I+1	GZC	810
116 IF (J.GT.N) GO TO 118	GZC	820
U2=REAL(EVALS(J))	GZC	830
U2=AIMAG(EVALS(J))	GZC	840
AU=ABS(U2-U1)	GZC	850
AU=ABS(U2-U1)	GZC	850
IF ((AU.LT.1.000E-08).AND.(AU.LT.1.00E-08)) IWARN1=1	GZC	870
J=J+1	GZC	880
GO TO 116	GZC	890
118 IF (ABS(U1).GT.1.00E-08) GO TO 120	GZC	900
IWARN2=1	GZC	910
EVALS(I)=CMPLX(-0.1000,U1)	GZC	920
120 CONTINUE	GZC	930
IF (IWARN1.EQ.1) WRITE (6,124)	GZC	940
IF (IWARN2.EQ.1) WRITE (6,126)	GZC	950
C	GZC	960
C*****WRITE THE INVERSE OF THE NODE ADMITTANCE IN PFE FORM, IF DESIRED	GZC	970
C	GZC	980
IF (PR.NE.1) RETURN	GZC	990
CALL GZOCPR (CH,BH,Y,N,NPORT,DMAT,EVALS,MP,MS)	GZC	1000
RETURN	GZC	1010
122 WRITE (6,128)	GZC	1020
RETURN	GZC	1030
C	GZC	1040
124 FORMAT (1H0,13H** WARNING **,1H ,47HREPEATED EIGENVALUES, ANSWERS	GZC	1050
1 MAY BE INACCURATE)	GZC	1060
126 FORMAT (1H0,13H** WARNING **,1H ,37HJW-AXIS POLE PRESENT HAS BEEN	GZC	1070
1 SHIFTED)	GZC	1080
128 FORMAT (1H0,22HSINGULAR MODAL MATRIX)	GZC	1090
C	GZC	1100
END	GZC	1110
SUBROUTINE GZOCPR (CHAT,BHAT,X,NSTU,NPORT,DMAT,EVALS,MP,MS)	GZP	10
C	GZP	20
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	GZP	30
C	GZP	40
C	GZP	50
C	GZP	60
C	GZP	70
C***** THIS SUB-PROGRAM IS GLOSSARY OF FORTRAN NAMES:	GZP	80
C	GZP	90
C	GZP	90

C	*	ZDC.	*GZP	100
C	*		*GZP	110
C	*****		*GZP	120
C		COMPLEX CHAT(MP,1),DMAT(MS,1),X(MS,1),EVALS(1)	GZP	130
		DIMENSION NPORT(1), DMAT(MP,1)	GZP	140
		COMMON /ENDS/ NCAP,NDVS,NRES,NIND,NDCS,NCPR	GZP	150
		WRITE (6,106)	GZP	160
		DO 104 I=1,NCPR	GZP	170
		DO 104 J=1,NCPR	GZP	180
		WRITE (6,108) NPORT(I),NPORT(J)	GZP	190
		DO 102 K=1,NSTU	GZP	200
102		X(1,K)=CHAT(I,K)*DMAT(K,J)	GZP	210
		WRITE (6,110) (X(1,K),EVALS(K),K=1,NSTU)	GZP	220
		WRITE (6,112) DMAT(I,J)	GZP	230
104		CONTINUE	GZP	240
		RETURN	GZP	250
C			GZP	260
		106 FORMAT (1H1,29HOPEN CIRCUIT IMPEDANCE MATRIX)	GZP	270
		108 FORMAT (1X,2HZ(,1X,1Z,1H,,1X,1Z,2H):,/1H ,12X,7HRESIDUE,27X,10HEIG	GZP	280
		109 VALUE)	GZP	290
		110 FORMAT (1H ,5X,E12.5,2H+J,E12.5,4X,E12.5,2H+J,E12.5)	GZP	300
		112 FORMAT (1H0,5HCONSTANT=,E12.5)	GZP	310
C			GZP	320
		END	GZP	330
		SUBROUTINE MORDR1 (NFREQ,NSTU,BHAT,CHAT,EU,H,N2FREQ,DMT,ZS,ZS1,LOSHO1	GZP	340
		1RC,NZT,NZT1,MIX,PHASE,MP,MS)	H01	10
C			H01	20
			H01	30
C	*****		H01	40
C	*		H01	50
C	*****	THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	H01	60
C	*	1. COMPUTE THE FIRST-ORDER TRANSFER FUNCTION AT EACH	H01	70
C	*	POSITIVE AND NEGATIVE INPUT FREQUENCY VALUE.	H01	80
C	*		H01	90
C	*****	THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	H01	100
C	*	NFREQ : NUMBER OF POSITIVE INPUT FREQUENCIES	H01	110
C	*	NSTU : NUMBER OF STATE VARIABLES (CIRCUIT COMPLEXITY)	H01	120
C	*	H(I,J) : I-TH PORT FIRST-ORDER TRANSFER FUNCTION VALUE	H01	130
C	*	AT H1(J) FREQUENCY VALUE	H01	140
C	*	ALL OTHER VARIABLE NAMES AND ARRAYS AS DEFINED IN	H01	150
C	*	SUB-PROGRAM AGAIN	H01	160
C	*		H01	170
C	*****		H01	180
C		COMPLEX SUM,S,EU(1),CHAT(MP,1),BHAT(MS,1),H(MP,1),TH	H01	190
		DIMENSION DMT(MP,1), NPORT(1)	H01	200
		COMMON /C03/ H1(10),AMP(10),TH(10),LUNIT	H01	210
		DIMENSION PHASE(5)	H01	220
		COMMON /016/ NCONT(32),JCONT(10)	H01	230
		COMMON /ENDS/ NCAP,NDVS,NRES,NIND,NDCS,NOUT	H01	240
C			H01	250
		***** FORM POSITIVE AND NEGATIVE FREQUENCY ARRAY FOR ANALYSIS	H01	260
C			H01	270
		DO 105 I=1,NFREQ	H01	280
		K=NFREQ+I	H01	290
		PHASE(I)=3.141592654*PHASE(I)/180.0000	H01	300
		TH(I)=CMPLX(0.0000,PHASE(I))	H01	310
		TH(K)=CMPLX(0.0000,-PHASE(I))	H01	320
		AMP(K)=AMP(I)	H01	330
105		H1(K)=-H1(I)	H01	340
			H01	350

```

C
C***** OBTAIN THE FIRST-ORDER TRANSFER FUNCTION AT EACH FREQUENCY POINT
C
      DO 140 L=1,NFREQ
        IF (LUNIT.EQ.3H HZ) W1=2.00000*3.141592654*W1(L)
        S=CMPLX(0.00,W1)
        DO 135 I=1,NOUT
          SUM=CMPLX(0.00,0.00)
          DO 110 K=1,NSTU
            SUM=SUM+CHAT(I,K)*DHAT(K,1)/(S-EU(K))
            DHAT=DMT(I,1)
            GO TO (115,120,125), NZT
          110 H(I,L)=(SUM+CMPLX(DHAT,0.0000))/CMPLX(ZS,0.0000)
            GO TO 130
          120 H(I,L)=(SUM+CMPLX(DHAT,0.0000))*ZS*S
            GO TO 130
          125 H(I,L)=(SUM+CMPLX(DHAT,0.0000))/ZS/S
          130 H(I,NFREQ+L)=CONJG(H(I,L))
          135 CONTINUE
        140 CONTINUE
C
C***** COMPUTE RESPONSE DUE TO SECOND-GENERATOR, IF PRESENT
C
      IF (MIX.NE.1) GO TO 175
      INP2=NCONT(LGSRG)
      N2FREQ=2*NFREQ
      DO 170 I=1,NOUT
        SUM=CMPLX(0.0000,0.0000)
        DO 145 K=1,NSTU
          SUM=SUM+CHAT(I,K)*DHAT(K,INP2)/(S-EU(K))
          DHAT=DMT(I,INP2)
          GO TO (150,155,160), NZT1
        145 H(I,NFREQ)=(SUM+CMPLX(DHAT,0.0000))/CMPLX(ZS1,0.0000)
          GO TO 165
        150 H(I,NFREQ)=(SUM+CMPLX(DHAT,0.0000))*ZS1*S
          GO TO 165
        160 H(I,NFREQ)=(SUM+CMPLX(DHAT,0.0000))/ZS1/S
        165 H(I,N2FREQ)=CONJG(H(I,NFREQ))
        170 CONTINUE
      175 RETURN
C
      END
      SUBROUTINE HORDR2 (NFREQ,NSTU,NNELEM,EU,DHAT,CHAT,H,H2,N2FREQ,W2,NAC2,
      1ICS,TEMP,DMT,N2FRPT,FCU,NPOUT,MP,MS)
C
C*****
C
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:
C
C      1. COMPUTE THE SECOND-ORDER TRANSFER FUNCTION VALUES AT
C      COMBINATION OF A PAIR OF POSITIVE AND NEGATIVE INPUT
C      FREQUENCY VALUES.
C
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:
C
C      H2(I,J) : I-TH PORT SECOND-ORDER TRANSFER FUNCTION
C               VALUE AT FREQUENCY W2(J)
C      W2(I)   : I-TH FREQUENCY VALUE APPEARING IN THE SECOND-
C               ORDER SPECTRUM
C      N2FRPT  : TOTAL NUMBER OF FREQUENCY POINTS APPEARING
C               IN THE SECOND-ORDER SPECTRUM

```

```

C      *      FCU(I)      : I-TH FREQUENCY COMBINATION CODE      *H02  180
C      *      SRC2(L)     : SECOND-ORDER CURRENT SOURCE DUE TO THE L-TH *H02  190
C      *                      NONLINEAR ELEMENT                  *H02  200
C      *      ALL OTHER VARIABLE NAMES AND ARRAYS AS DEFINED IN  *H02  210
C      *      SUB-PROGRAM AMASH                                     *H02  220
C      *                                                                *H02  230
C*****                                                                *H02  240
C      INTEGER FCU(1)                                             *H02  250
C      COMPLEX SUM,S,EU(1),CHAT(NS,1),CHAT(MP,1),H(MP,1),H2(MP,1) *H02  260
C      COMPLEX SRC2(25),TEMP(MP,1),CP,SD,SS,TH                  *H02  270
C      DIMENSION H2(1), HNT(MP,1)                                *H02  280
C      COMMON /801/ NTYPE(10),AI(10,9)                          *H02  290
C      COMMON /802/ H1(10),GMP(10),TH(10),LUNIT                 *H02  300
C      COMMON /803/ NSCNT(92),JCNT(10)                           *H02  310
C      COMMON /804/ NDCS,NDUS,NRES,NEND,NDCS,NCS                 *H02  320
C      DATA NL/2HNL/                                             *H02  330
C      K=0                                                         *H02  340
C      NFREQ=2*NFREQ                                              *H02  350
C                                                                *H02  360
C                                                                *H02  370
C*****INITIALIZE                                                *H02  380
C      DO 105 I=1,NDCS                                           *H02  390
C      105 SRC2(I)=CMPLX(0.00,0.00)                               *H02  400
C                                                                *H02  410
C***** COMPUTE SECOND-ORDER TRANSFER FUNCTIONS AT EACH FREQUENCY COMBI *H02  420
C      DO 155 II=1,NCFREQ                                         *H02  430
C      155 JJ=II,NCFREQ                                           *H02  440
C      SUM=H1(II)+H1(JJ)                                          *H02  450
C      K=K+1                                                       *H02  460
C      H2(K)=SUM                                                   *H02  470
C      FCU(K)=10*II+JJ                                             *H02  480
C      IF (LUNIT.EQ.2H H2) SUM=2.000000*3.141592654*(W1(II)+W1(JJ)) *H02  490
C      S=CMPLX(0.00,SUM)                                           *H02  500
C                                                                *H02  510
C                                                                *H02  520
C                                                                *H02  530
C***** FORM SECOND-ORDER CURRENT SOURCE VECTOR                  *H02  540
C      DO 130 L=1,NNELEM                                           *H02  550
C      LS=9*(L-1)+NFCUT+2                                          *H02  560
C      ICON1=NCNT(L3)                                              *H02  570
C      ICON2=NCNT(L3+1)                                            *H02  580
C      INDEX=JCNT(L)                                               *H02  590
C      GO TO (110,115,120,125), INDEX                             *H02  600
C                                                                *H02  610
C                                                                *H02  620
C***** NONLINEAR CAPACITIVE SOURCE                              *H02  630
C      110      SRC2(L)=H(ICON1,II)*H(ICON2,JJ)*AI(L,2)*S        *H02  640
C      GO TO 130                                                  *H02  650
C                                                                *H02  660
C***** NONLINEAR INDUCTIVE SOURCE                               *H02  670
C      115      IF (SUM.EQ.0.00) GO TO 130                         *H02  680
C      SRC2(L)=H(ICON1,II)*H(ICON2,JJ)*AI(L,2)/S                *H02  690
C      GO TO 130                                                  *H02  700
C                                                                *H02  710
C                                                                *H02  720
C***** NONLINEAR DEPENDENT SOURCE                              *H02  730
C      120      SP=H(ICON1,II)*H(ICON1,JJ)*AI(L,3)               *H02  740
C      SD=H(ICON2,II)*H(ICON2,JJ)*AI(L,4)                       *H02  750
C                                                                *H02  760
C                                                                *H02  770

```

```

1      SS=(H(ICON1,II)*H(ICON2,JJ)+H(ICON2,II)*H(ICON1,JJ))*AI(L,5)H02 780
      /2.00 H02 790
      SRC2(L)=+(SP+SQ+SS) H02 800
      GO TO 130 H02 810
C H02 820
C***** NONLINEAR RESISTIVE SOURCE H02 830
C H02 840
125      SRC2(L)=H(ICON1,II)*H(ICON2,JJ)*AI(L,2) H02 850
130      CONTINUE H02 860
C H02 870
C***** FORM ZOC( S1+S2 ) H02 880
C H02 890
      DO 140 J=1,NCS H02 900
      DO 140 M=1,NCS H02 910
      SUM=CMPLX(0.00,0.00) H02 920
      DO 135 L=1,NSTU H02 930
135      SUM=SUM+CHAT(J,L)*BHAT(L,M)/(S-EU(L)) H02 940
      DHAT=DMT(J,M) H02 950
      TEMP(J,M)=SUM+CMPLX(DHAT,0.0000) H02 960
140      CONTINUE H02 970
C H02 980
C***** OBTAIN SECOND-ORDER TRANSFER FUNCTIONS H02 990
C H02 1000
      DO 150 J=1,NCS H02 1010
      SUM=CMPLX(0.00,0.00) H02 1020
      DO 145 M=1,NNELEM H02 1030
      M3=3*(M-1)+NPOUT+1 H02 1040
      ICON=NCONT(M3) H02 1050
145      SUM=SUM+TEMP(J,ICON)*SRC2(M) H02 1060
      H2(J,K)=SUM H02 1070
      IF ((NTYPE(J).EQ.NL).AND.(SUM.EQ.00.00)) H2(J,K)=0.00 H02 1080
150      CONTINUE H02 1090
C H02 1100
155      CONTINUE H02 1110
      N2FRPT=K H02 1120
      RETURN H02 1130
C H02 1140
      END H02 1150
      SUBROUTINE HORDR3 (NFR,NSTU,NNELEM,EU,DHAT,CHAT,H1,H2,N2F,W3,H3,NH02 1160
1CS,TEMP,DMT,KK,FCU,NPOUT,MP,MS) H02 1170
C H02 1180
C***** H02 1190
C * H02 1200
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION: H02 1210
C * 1. COMPUTE THE THIRD-ORDER TRANSFER FUNCTION VALUES AT H02 1220
C * EACH POSITIVE COMBINATION OF THREE POSITIVE AND H02 1230
C * NEGATIVE INPUT FREQUENCIES TAKEN AT A TIME. H02 1240
C * H02 1250
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES: H02 1260
C * H3(I,J) : I-TH PORT THIRD-ORDER TRANSFER FUNCTION VALUE H02 1270
C * AT FREQUENCY W3(J) H02 1280
C * W3(J) : J-TH POSITIVE FREQUENCY VALUE APPEARING IN H02 1290
C * THE THRD-ORDER SPECTRUM H02 1300
C * FCU(J) : W3(J) FREQUENCY COMBINATION CODE H02 1310
C * SRC3(L) : THIRD-ORDER CURRENT SOURCE DUE TO L-TH H02 1320
C * NONLINEAR ELEMENT H02 1330
C * ALL OTHER VARIABLE NAMES AND ARRAYS AS DEFINED IN H02 1340
C * SUB-PROGRAM AMAIN H02 1350
C * H02 1360
C***** H02 1370

```

C	INTEGER FCU(1)	H03	230
	COMPLEX SUM, S, EU(1), EHAT(MS,1), CHAT(MP,1), H1(MP,1), H2(MP,1), SRC3(2	H03	240
	15), TEMP(MP,1), H3(MP,1), G31, G32, G33, G34, G231, G232, G233, G234, TH	H03	250
	DIMENSION H3(1), DMT(MP,1)	H03	260
	COMMON /001/ NTYPE(10), AI(10,9)	H03	270
	COMMON /003/ W1(10), AMP(10), TH(10), LUNIT	H03	280
	COMMON /016/ NCONT(32), JCONT(10)	H03	290
	COMMON /ENDS/ NCAP, NDVS, NRES, NEND, NDCS, NCS	H03	300
	DATA NL/2XNL/	H03	310
	KK=0	H03	320
C		H03	330
C*****	INITIALIZE	H03	340
C		H03	350
	DO 105 I=1, NDCS	H03	360
	105 SRC3(I)=CMPLX(0.00, 0.00)	H03	370
C		H03	380
C*****	COMPUTE THIRD-ORDER TRANSFER FUNCTION AT EACH FREQUENCY COMB	H03	390
C		H03	400
	DO 155 I=1, NFR	H03	410
	DO 155 J=1, NEF	H03	420
	DO 155 K=J, NEF	H03	430
	DUM=H1(I)+H1(J)+H1(K)	H03	440
	IF (DUM.LT.0.00) GO TO 155	H03	450
	KK=KK+1	H03	460
	H3(KK)=DUM	H03	470
	FCU(KK)=100*I+10*J+K	H03	480
	IF (LUNIT.EQ.24 HZ) DUM=2.0000*3.141592654*DUM	H03	490
	S=CMPLX(0.00, DUM)	H03	500
	IDUM=(I-1)*N2F-I*(I-1)/2	H03	510
	I1=IDUM+J	H03	520
	I2=IDUM+K	H03	530
	J1=(J-1)*N2F-J*(J-1)/2+K	H03	540
C		H03	550
C*****	FORM NONLINEAR CURRENT SOURCE VECTOR	H03	560
C		H03	570
	DO 130 L=1, NNELEM	H03	580
	L3=3*(L-1)+NPOUT+2	H03	590
	IC1=NCONT(L3)	H03	600
	IC2=NCONT(L3+1)	H03	610
	G31=H1(IC1, I)*H1(IC1, J)*H1(IC1, K)	H03	620
	G231=H1(IC1, I)*H2(IC1, J1)+H1(IC1, J)*H2(IC1, I2)+H1(IC1, K)*H2(IC1, K)	H03	630
	1 IC1, I1)	H03	640
	G231=2.0000*G231/3.00000	H03	650
	INDEX=JCONT(L)	H03	660
	GO TO (110, 115, 120, 123), INDEX	H03	670
C		H03	680
C*****	NONLINEAR CAPACITIVE SOURCE	H03	690
C		H03	700
	110 SRC3(L)=(G31*AI(L,3)+G231*AI(L,2))*S	H03	710
	GO TO 130	H03	720
C		H03	730
C*****	NONLINEAR INDUCTIVE SOURCE	H03	740
C		H03	750
	115 IF (DUM.EQ.0.00) GO TO 130	H03	760
	SRC3(L)=(G31*AI(L,3)+G231*AI(L,2))/S	H03	770
	GO TO 130	H03	780
C		H03	790
C*****	NONLINEAR DEPENDENT SOURCE	H03	800
C		H03	810
		H03	820

```

120      G232=H1(IC2,I)*H2(IC2,J1)+H1(IC2,J)*H2(IC2,I2)+H1(IC2,K)*H2(H03 830
1      IC2,I1)                                     H03 840
      G233=H1(IC1,I)*H2(IC2,J1)+H1(IC1,J)*H2(IC2,I2)+H1(IC1,K)*H2(H03 850
1      IC2,I1)                                     H03 860
      G234=H1(IC2,I)*H2(IC1,J1)+H1(IC2,J)*H2(IC1,I2)+H1(IC2,K)*H2(H03 870
1      IC1,I1)                                     H03 880
      G32=H1(IC2,I)*H1(IC2,J)*H1(IC2,K)*AI(L,7)      H03 890
      G33=(H1(IC1,I)*H1(IC1,J)*H1(IC2,K)+H1(IC1,J)*H1(IC1,K)*H1(IC1,H03 900
1      2,I)+H1(IC1,K)*H1(IC1,I)*H1(IC2,J))*AI(L,8)/3.0000      H03 910
      G34=(H1(IC1,I)*H1(IC2,J)*H1(IC2,K)+H1(IC1,J)*H1(IC2,K)*H1(IC1,H03 920
1      2,I)+H1(IC1,K)*H1(IC2,I)*H1(IC2,J))*AI(L,9)/3.000      H03 930
      G231=G231*AI(L,3)      H03 940
      G232=2.0000*G232*AI(L,4)/3.00000      H03 950
      SRC3(L)=G231+G232+(G233+G234)*AI(L,5)/3.00+G31*AI(L,6)+G32+G33 960
1      33+G24      H03 970
      GO TO 130      H03 980
C      H03 990
C***** NONLINEAR RESISTIVE SOURCE      H03 1000
C      H03 1010
125      SRC3(L)=G31*AI(L,3)+G231*AI(L,2)      H03 1020
130      CONTINUE      H03 1030
C      H03 1040
C***** FORM ZOC ( S1+S2+S3 )      H03 1050
C      H03 1060
      DO 140 JJ=1,NCS      H03 1070
      DO 140 M=1,NCS      H03 1080
      SUM=CMPLX(0.00,0.00)      H03 1090
      DO 135 L=1,NSTU      H03 1100
135      SUM=SUM+CHAT(JJ,L)*BHAT(L,M)/(S-EV(L))      H03 1110
      DHAT=DMT(JJ,M)      H03 1120
      TEMP(JJ,M)=SUM+CMPLX(DHAT,0.000)      H03 1130
140      CONTINUE      H03 1140
      DO 150 JJ=1,NCS      H03 1150
      SUM=CMPLX(0.00,0.00)      H03 1160
      DO 145 M=1,NNELEM      H03 1170
      M3=3*(M-1)+NPOUT+1      H03 1180
      ICON=NCONT(M3)      H03 1190
145      SUM=SUM+TEMP(JJ,ICON)*SRC3(M)      H03 1200
      H3(JJ,KK)=SUM      H03 1210
      IF ((NTYPE(JJ).EQ.NL).AND.(DUM.EQ.0.00)) H3(JJ,KK)=0.00      H03 1220
150      CONTINUE      H03 1230
C      H03 1240
155 CONTINUE      H03 1250
      RETURN      H03 1260
C      H03 1270
      END      H03 1280
      SUBROUTINE IWRIST (NFREQ,H1,NPORT,IAP,NOUT,MP)      IW1 10
C      IW1 20
C*****      IW1 30
C      *      IW1 40
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:      IW1 50
C      *      1. PRINT THE FIRST-ORDER TRANSFER FUNCTION AND OUTPUT      IW1 60
C      *      VOLTAGE VALUES AT EACH POSITIVE INPUT FREQUENCY VALUE      IW1 70
C      *      AT THE REQUESTED PORTS.      IW1 80
C      *      IW1 90
C***** THIS SUB-PROGRAM#S GLOSSARY OF FORTRAN NAMES:      IW1 100
C      *      H1(I,J) : UPON ENTRANCE: I-TH PORT FIRST-ORDER TRANSFER      IW1 110
C      *      FUNCTION VALUE AT W1(J); UPON EXIT: I-TH PORT      IW1 120
C      *      OUTPUT VOLTAGE VALUE AT FREQUENCY W1(J)      IW1 130
C      *      IAP : PRINTING OPTION FLAG VARIABLE      IW1 140

```

C	*	ALL OTHER VARIABLES NAMES AND ARRAYS AS DEFINED IN	*IW1	150
C	*	SUB-PROGRAM AMAIN.	*IW1	160
C	*		*IW1	170
C	*****		*IW1	180
C			IW1	190
		DIMENSION NPORT(1)	IW1	200
		COMPLEX H1(KP,1),TH	IW1	210
		COMMON /003/ W1(10),AMP(10),TH(10),LUNIT	IW1	220
		COMMON /ENDS/ NCAP,NDUS,NRES,NIND,NDCS,NCS	IW1	230
C			IW1	240
C	*****	CHECK IF RESPONSE IS TO BE PRINTED AT ALL EXTRACTED PORTS	IW1	250
C			IW1	260
		IF (IAP.EQ.1) GO TO 105	IW1	270
		L=2	IW1	280
		K=NOUT	IW1	290
		GO TO 110	IW1	300
	105	L=1	IW1	310
		K=NCS	IW1	320
C			IW1	330
C	*****	PRINT THE FIRST-ORDER TRANSFER FUNCTION AND RESPONSE AT EACH	IW1	340
C	*****	OUTPUT PORT AND FREQUENCY	IW1	350
C			IW1	360
	110	DO 130 I=1,NFREQ	IW1	370
		WRITE (6,135) I,W1(I),LUNIT	IW1	380
		WRITE (6,140)	IW1	390
		WRITE (6,145)	IW1	400
		DO 125 J=L,K	IW1	410
		IOUT=NPORT(J)	IW1	420
		AMAGN=CABS(H1(IOUT,I))	IW1	430
		U=-REAL(H1(IOUT,I))	IW1	440
		V=-AIMAG(H1(IOUT,I))	IW1	450
		H1(IOUT,I)=AMP(I)*H1(IOUT,I)*CEXP(TH(I))	IW1	460
		YMAC=CABS(H1(IOUT,I))	IW1	470
		YU=-REAL(H1(IOUT,I))	IW1	480
		YV=-AIMAG(H1(IOUT,I))	IW1	490
		IF (AMAGN.EQ.0.0000) GO TO 115	IW1	500
		ADB=20.000*ALOG10(AMAGN)	IW1	510
		PHASE=ATAN2(YU,YV)*180.000/3.141592654	IW1	520
		GO TO 120	IW1	530
	115	ADB=-1.00E+30	IW1	540
		PHASE=0.00000	IW1	550
	120	WRITE (6,150) IOUT,U,V,AMAGN,ADB,YU,YV,YMAC,PHASE	IW1	560
	125	CONTINUE	IW1	570
	130	CONTINUE	IW1	580
		RETURN	IW1	590
C			IW1	600
	135	FORMAT (1H0,12HFIRST ORDER: ,15X,11HFREQUENCY(,11,5H)= ,E10.3,2X	IW1	610
		1,AS)	IW1	620
	140	FORMAT (1H0,34X,17HTRANSFER FUNCTION,40X,14HOUTPUT VOLTAGE)	IW1	630
	145	FORMAT (//1X,9H PORT ,8X,4HREAL,8X,8HIMAGINARY,6X,9HMAGNITUDE,6X,9H	IW1	640
		1X,SH20LOG MAG,8X,4HREAL,8X,8HIMAGINARY,6X,9HMAGNITUDE,6X,5HPHASE, /	IW1	650
		2,5X,2HND,114X,SHDEC,/,1H ,3X,4(1H.),6X,57(1H.),3X,53(1H.))	IW1	660
	150	FORMAT (1H ,4X,I2,4X,7(3X,E12.5),3X,F7.2)	IW1	670
C			IW1	680
		END	IW1	690
		SUBROUTINE IWR2ND (NFREQ,N2FRPT,H2,NPORT,I2FC,W2,IAP,NOUT,MP)	IW2	10
C			IW2	20
C	*****		IW2	30
C	*		*IW2	40
C	*****	THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	*IW2	50


```

      GO TO 120
115   ADD=-1.000E+30
      PHASE=0.0000
120   WRITE (G,150) ICUT,U,U,AMAGN,AD3,YU,YU,YMAG,PHASE
125   CONTINUE
130   CONTINUE
      RETURN
C
135   FORMAT (1H0,16HSECOND ORDER: ,16X,11HFREQUENCY( ,11,1H,,12,5H )= ,1W2 630
      1E10.3,2X,49)
140   FORMAT (1H0,34X,17HTRANSFER FUNCTION,40X,14HOUTPUT VOLTAGE)
145   FORMAT (/1H,5H   PORT ,6X,4HREAL,6X,5HIMAGINARY,6X,5HMAGNITUDE,6W2 640
      1X,5H20LCB 1H3,6X,4HREAL,6X,5HIMAGINARY,6X,5HMAGNITUDE,6X,5H5PHASE, /1W2 650
      2,6X,2HNO,11H,5H25, /1H ,6X,4H1H,1,6X,5H1H.,6X,5H1H.)
150   FORMAT (1H ,4X,20,4X,7X,5H12.5),2X,F7.2)
      1W2 660
C
      END
      SUBROUTINE INTRRD (NFREQ,NSEPT,NO,NPORT,ISFC,W3,IAP,NCUT,MP)
      1W2 670
      1W2 680
C
C*****
      1W2 690
C
      1W2 700
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:
      1W2 710
C
      1. PRINT THE THIRD-ORDER TRANSFER FUNCTION AND OUTPUT
      1W2 720
C
      VOLTAGE VALUES AT EACH OF THE NON-NEGATIVE FREQUENCY
      1W2 730
C
      VALUES AT THE REQUESTED OUTPUT PORTS.
      1W2 740
C
C***** THIS SUB-PROGRAMS GLOSSARY OF FORTRAN NAMES:
      1W2 750
C
      NSEPT : TOTAL NUMBER OF NON-NEGATIVE FREQUENCY
      1W2 760
C
      COMPONENTS IN THE THIRD-ORDER SPECTRUM
      1W2 770
C
      H3(I,J) : UPON ENTRANCE: I-TH PORT THIRD-ORDER TRANSFER=1W2 780
      FUNCTION VALUE AT H3(I,J); UPON EXIT: I-TH PORT=1W2 790
C
      THIRD-ORDER OUTPUT VOLTAGE AT FREQUENCY W3(I,J)=1W2 800
C
      ISFC(J) : W3(J) FREQUENCY COMBINATION CODE
      1W2 810
C
      ALL OTHER VARIABLE NAMES AND ARRAYS AS DEFINED IN
      1W2 820
C
      SUB-PROGRAM MAIN
      1W2 830
C
C*****
      1W2 840
C
      DIMENSION NPORT(1), ISFC(1), W3(1)
      1W2 850
C
      COMPLEX H3(NP,1),TH
      1W2 860
C
      COMMON /C03/ W1(10),AMP(10),TH(10),LUNIT
      1W2 870
C
      COMMON /ENDS/ NCAP,NCUS,NRES,NEND,NDCS,NCS
      1W2 880
C
C***** CHECK IF RESPONSE IS TO BE PRINTED FOR ALL EXTRACTED PORTS
      1W2 890
C
      IF (IAP.EQ.1) GO TO 105
      1W2 900
C
      K=2
      1W2 910
C
      L=NCUT
      1W2 920
C
      GO TO 110
      1W2 930
C
      105 K=1
      1W2 940
C
      L=NCS
      1W2 950
C
C***** PRINT THIRD-ORDER TRANSFER FUNCTION AND RESPONSE AT EACH OUTPUT
      1W2 960
C
      C***** PORT AND POSITIVE FREQUENCY POINT
      1W2 970
C
      110 DO 150 I=1,NSEPT
      1W2 980
C
C***** DECIPIER FREQUENCY COMBINATION
      1W2 990
C
      ICOND=ISFC(I)
      1W2 1000

```

II=ICOMB/100	IW3	440
JJ=(ICOMB-100*II)/10	IW3	450
KK=ICOMB-100*II-10*JJ	IW3	450
J1=JJ	IW3	470
K1=KK	IW3	480
IF (J1.GT.NFREQ) J1=NFREQ-J1	IW3	490
IF (K1.GT.NFREQ) K1=NFREQ-K1	IW3	500
WRITE (6,133) II,J1,K1,W3(I),LUNIT	IW3	510
WRITE (6,140)	IW3	520
WRITE (6,145)	IW3	530
DO 125 J=K,L	IW3	540
IOUT=NPORT(J)	IW3	550
AMAGN=CABS(H3(IOUT,I))	IW3	550
U=REAL(H3(IOUT,I))	IW3	570
V=AIMAG(H3(IOUT,I))	IW3	580
DIV=CMPLX(1.00,0.00)	IW3	590
IF ((II.EQ.JJ).OR.(JJ.EQ.KK)) DIV=CMPLX(2.000,00.00)	IW3	600
IF ((II.EQ.JJ).AND.(JJ.EQ.KK)) DIV=CMPLX(6.00,0.00)	IW3	610
H3(IOUT,I)=1.500*AMP(II)*AMP(JJ)*AMP(KK)*H3(IOUT,I)/DIV	IW3	620
H3(IOUT,I)=H3(IOUT,I)*CEXP(TH(II))*CEXP(TH(JJ))*CEXP(TH(KK))	IW3	630
YMAG=CABS(H3(IOUT,I))	IW3	640
YU=REAL(H3(IOUT,I))	IW3	650
YV=AIMAG(H3(IOUT,I))	IW3	650
IF (AMAGN.EQ.0.0000) GO TO 115	IW3	670
ADB=20.000*ALOG10(AMAGN)	IW3	680
PHASE=ATAN2(YV,YU)*180.00/3.141592654	IW3	690
GO TO 120	IW3	700
115 ADB=-1.000E+30	IW3	710
PHASE=0.000	IW3	720
120 WRITE (6,150) IOUT,U,V,AMAGN,ADB,YU,YV,YMAG,PHASE	IW3	730
125 CONTINUE	IW3	740
130 CONTINUE	IW3	750
RETURN	IW3	760
C	IW3	770
135 FORMAT (1H0,12HTHIRD ORDER:,15X,11HFREQUENCY(,11,1H,,12,1H,,12,5HIN3	IW3	780
1 1)=,E10.3,2X,A3)	IW3	790
140 FORMAT (1H0,34X,17HTRANSFER FUNCTION,40X,14HOUTPUT VOLTAGE)	IW3	800
145 FORMAT (/1X,9H PORT ,8X,4HREAL,8X,9HIMAGINARY,6X,9HMAGNITUDE,6H1X3	IW3	810
1X,9H20LOG MAG,9X,4HREAL,8X,9HIMAGINARY,6X,9HMAGNITUDE,6X,5HPHASE,1W3	IW3	820
2,5X,2HNO,114X,3HDEG,/,1H ,3X,4(1H.),6X,57(1H.),3X,53(1H.))	IW3	830
150 FORMAT (1H ,4X,12,4X,7(3X,E12.5),3X,F7.2)	IW3	840
C	IW3	850
END	IW3	860
SUBROUTINE JSPCTM (Y1,Y2,Y3,NFREQ,N2FRPT,N3FRPT,FR,Y,MOUT,W2,W3,IPJSP	IW3	10
IT,YLG,MP)	JSP	20
C	JSP	30
C*****	JSP	40
C	JSP	50
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTIONS:	JSP	60
C	JSP	70
C	JSP	80
C	JSP	90
C	JSP	100
C***** THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINES:	JSP	110
C	JSP	120
C	JSP	130
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	JSP	140
C	JSP	150
C	JSP	160
C	JSP	170

```

C      *      NFREQ      : TOTAL NUMBER OF INPUT FREQUENCIES      *JSP 180
C      *      N2FRPT      : TOTAL NUMBER OF POSITIVE AND NEGATIVE *JSP 190
C      *      NSFRPT      : TOTAL NUMBER OF NON-NEGATIVE FREQUENCIES *JSP 200
C      *      FR          : VALUES OF DISTINCT FREQUENCIES IN THE OUTPUT *JSP 210
C      *      Y(I)        : OUTPUT PORT VOLTAGE AT FREQUENCY FR(I) *JSP 220
C      *      NOUT        : OUTPUT PORT INDEX *JSP 230
C      *      YLG(I)      : LOG OF THE OUTPUT VOLTAGE AT FREQUENCY FR(I) *JSP 240
C      *      *          *          *          *          *          *JSP 250
C      *          *          *          *          *          *JSP 260
C      *          *          *          *          *          *JSP 270
C      *          *          *          *          *          *JSP 280
C      *          *          *          *          *          *JSP 290
C      *          *          *          *          *          *JSP 300
C      *          *          *          *          *          *JSP 310
C      *          *          *          *          *          *JSP 320
C      *          *          *          *          *          *JSP 330
C      *          *          *          *          *          *JSP 340
C      *          *          *          *          *          *JSP 350
C      *          *          *          *          *          *JSP 360
C      *          *          *          *          *          *JSP 370
C      *          *          *          *          *          *JSP 380
C      *          *          *          *          *          *JSP 390
C      *          *          *          *          *          *JSP 400
C      *          *          *          *          *          *JSP 410
C      *          *          *          *          *          *JSP 420
C      *          *          *          *          *          *JSP 430
C      *          *          *          *          *          *JSP 440
C      *          *          *          *          *          *JSP 450
C      *          *          *          *          *          *JSP 460
C      *          *          *          *          *          *JSP 470
C      *          *          *          *          *          *JSP 480
C      *          *          *          *          *          *JSP 490
C      *          *          *          *          *          *JSP 500
C      *          *          *          *          *          *JSP 510
C      *          *          *          *          *          *JSP 520
C      *          *          *          *          *          *JSP 530
C      *          *          *          *          *          *JSP 540
C      *          *          *          *          *          *JSP 550
C      *          *          *          *          *          *JSP 560
C      *          *          *          *          *          *JSP 570
C      *          *          *          *          *          *JSP 580
C      *          *          *          *          *          *JSP 590
C      *          *          *          *          *          *JSP 600
C      *          *          *          *          *          *JSP 610
C      *          *          *          *          *          *JSP 620
C      *          *          *          *          *          *JSP 630
C      *          *          *          *          *          *JSP 640
C      *          *          *          *          *          *JSP 650
C      *          *          *          *          *          *JSP 660
C      *          *          *          *          *          *JSP 670
C      *          *          *          *          *          *JSP 680
C      *          *          *          *          *          *JSP 690
C      *          *          *          *          *          *JSP 700
C      *          *          *          *          *          *JSP 710
C      *          *          *          *          *          *JSP 720
C      *          *          *          *          *          *JSP 730
C      *          *          *          *          *          *JSP 740
C      *          *          *          *          *          *JSP 750
C      *          *          *          *          *          *JSP 760
C      *          *          *          *          *          *JSP 770

```

IF (FR(J).EQ.PFREQ) GO TO 125	JSP	780
GO TO 130	JSP	790
125 Y(I)=Y(I)+Y(J)	JSP	800
IPT(J)=IPTVAL	JSP	810
130 CONTINUE	JSP	820
135 CONTINUE	JSP	830
C	JSP	840
C***** PRINT COMPLETE OUTPUT SPECTRUM	JSP	850
C	JSP	860
IF (LUNIT.EQ.3HRAD) ICON=1	JSP	870
WRITE (6,155) JOUT	JSP	880
WRITE (6,160) IFRUNT(ICON)	JSP	890
DO 150 I=1,KOUNT	JSP	900
IPTVAL=IPT(I)	JSP	910
IF (IPTVAL.LT.I) GO TO 150	JSP	920
AMAGN=CABS(Y(I))	JSP	930
U=REAL(Y(I))	JSP	940
U=AIMAG(Y(I))	JSP	950
IF (AMAGN.EQ.0.000) GO TO 140	JSP	960
YLG(I)=ALOG10(AMAGN)	JSP	970
PHASE=ATAN2(U,U)*180.000/3.141592654	JSP	980
GO TO 145	JSP	990
140 PHASE=0.000	JSP	1000
YLG(I)=-1.000E+30	JSP	1010
145 WRITE (6,165) FR(I),U,U,AMAGN,PHASE	JSP	1020
150 CONTINUE	JSP	1030
C	JSP	1040
C***** PLOT THE OUTPUT SPECTRUM	JSP	1050
C	JSP	1060
WRITE (6,170)	JSP	1070
CALL JPLTSP (FR,YLG,KOUNT,23,23HLOG OF OUTPUT MAGNITUDE)	JSP	1080
WRITE (6,175) IFRUNT(ICON)	JSP	1090
RETURN	JSP	1100
C	JSP	1110
155 FORMAT (1H1,/,1X,47HSINUSOIDAL STEADY-STATE OUTPUT RESPONSE AT PO	JSP	1120
1RT,2X,I2,/,1H,47(1H.))	JSP	1130
160 FORMAT (//7X,9HFREQUENCY,11X,4HREAL,12X,9HIMAGINARY,8X,9HMAGNITUDE	JSP	1140
1,12X,5HPHASE,7X,A7,67X,3HDEG,1H,6X,82(1H.))	JSP	1150
165 FORMAT (1H,5X,E12.2,4X,E12.3,7X,E12.3,5X,E12.3,7X,E12.3)	JSP	1160
170 FORMAT (1H1,45X,31HRESPONSE MAGNITUDE VS FREQUENCY/)	JSP	1170
175 FORMAT (1H0,55X,11HFREQUENCY (,A7,1H))	JSP	1180
C	JSP	1190
END	JSP	1200
SUBROUTINE JPLTSP (XX,YY,NDATA,NB,LABEL2)	JPT	10
C	JPT	20
C*****	JPT	30
C	JPT	40
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION:	JPT	50
C	JPT	60
C	JPT	70
C***** THIS SUB-PROGRAM USES THE FOLLOWING SUBROUTINES:	JPT	80
C	JPT	90
C	JPT	100
C	JPT	110
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	JPT	120
C	JPT	130
C	JPT	140
C	JPT	150
C	JPT	160
C	JPT	170

C	*	*JPT	130
C	*****	JPT	190
C	REAL JPUT	JPT	200
	DIMENSION SYMBOL(4), AMASK(10), XSCALE(12), YSCALE(51), PLOT(51,10)	JPT	210
	1)	JPT	220
	DIMENSION XX(1), YY(1)	JPT	230
	DIMENSION LABEL2(6), LB(60)	JPT	240
	DATA (SYMBOL(I), I=1,4)/10H*****10H0000000000,10H\$\$\$\$\$\$\$\$\$,1JPT	250	
	10HXXXXXXXXXX/, (AMASK(I), I=1,10)/770000000000000000003,770000000000JPT	260	
	2000000B,77000000000000003,770000000000003,770000000000B,7700000000JPT	270	
	3B,77000000B,770000B,7700B,77B/, BLANK/555555555555555555B/, DASH/1JPT	280	
	40H-----/, UPLINE/10HIIIIIIIIII/, PLUS/10H+++++++/	290	
		JPT	300
C	GENERATE GRAPH LABELS	JPT	310
C		JPT	320
C	N2=N3	JPT	330
	CALL JSEP (6,60,N2,LABEL2,LB)	JPT	340
C		JPT	350
C	ZERO GRAPH ARRAY TO ALL BLANKS	JPT	360
C		JPT	370
	DO 105 I=1,51	JPT	380
	DO 105 J=1,10	JPT	390
	105 PLOT(I,J)=BLANK	JPT	400
C		JPT	410
C	FIND DATA MAXIMUM AND MINIMUMS	JPT	420
C		JPT	430
	YMAX=XX(1)	JPT	440
	XMIN=XMAX	JPT	450
	YMAX=YY(1)	JPT	460
	YMIN=YMAX	JPT	470
	IF (NDATA.LE.0) GO TO 115	JPT	480
	DO 110 J=1,NDATA	JPT	490
	XMAX=AMAX1(XMAX,XX(J))	JPT	500
	XMIN=AMIN1(XMIN,XX(J))	JPT	510
	YMAX=AMAX1(YMAX,YY(J))	JPT	520
	YMIN=AMIN1(YMIN,YY(J))	JPT	530
	110 YMIN=AMIN1(YMIN,YY(J))	JPT	540
	115 CONTINUE	JPT	550
C		JPT	560
C	DETERMINE X AND Y INCREMENTS	JPT	570
C		JPT	580
	XSC=100./(XMAX-XMIN)	JPT	590
	YSC=50.0/(YMAX-YMIN)	JPT	600
C		JPT	610
C	CONSTRUCT HORIZONTAL REFERENCE LINES	JPT	620
C		JPT	630
	DO 125 I=1,51,10	JPT	640
	IF (I.LT.2) GO TO 125	JPT	650
	DO 120 J=1,10	JPT	660
	PLOT(I,J)=DASH	JPT	670
	120	JPT	680
	125 CONTINUE	JPT	690
C		JPT	700
C	CONSTRUCT VERTICAL REFERENCE LINE	JPT	710
C		JPT	720
	JPOS=10	JPT	730
	JWORD=10	JPT	740
	DO 130 I=1,51	JPT	750
	SYM=UPLINE	JPT	760
	TEST=(PLOT(I,JWORD).AND.AMASK(JPOS))	JPT	770
	TDASH=(DASH.AND.AMASK(JPOS))	JPT	780

	IF (JLCOMP(TEST,TDASH).EQ.0) SYM=PLUS	JPT 780
130	PLOT(I,JWORD)=JPUT(AMASK,I,JWORD,JPOS,SYM,S!,PLOT)	JPT 790
C		JPT 800
C	DETERMINE X,Y LOCATION OF DATA POINTS ON GRAPH	JPT 810
C		JPT 820
	NDF=NDATA	JPT 830
	DO 145 J=1,NDF	JPT 840
	JYKK=((YY(J)-YMIN)*YSC+1.5)	JPT 850
	JX=((XX(J)-XMIN)*XSC+0.5)	JPT 860
	JWORD=(JX/10)+1	JPT 870
	JPOS=MOD(JX,10)+1	JPT 880
	DO 145 JY=1,JYKK	JPT 890
	TEST=(PLOT(JY,JWORD).AND.AMASK(JPOS))	JPT 900
	TUP=(UPLINE.AND.AMASK(JPOS))	JPT 910
	TDLANK=(DLANK.AND.AMASK(JPOS))	JPT 920
	TDASH=(DASH.AND.AMASK(JPOS))	JPT 930
	TPLUS=(PLUS.AND.AMASK(JPOS))	JPT 940
	IF (JLCOMP(TDLANK,TEST).EQ.0) GO TO 135	JPT 950
	IF (JLCOMP(TDASH,TEST).EQ.0) GO TO 135	JPT 960
	IF (JLCOMP(TPLUS,TEST).EQ.0) GO TO 135	JPT 970
	IF (JLCOMP(TUP,TEST).NE.0) GO TO 140	JPT 980
C		JPT 990
C	INSERT SYMBOL FOR DATA POINT	JPT 1000
C		JPT 1010
135	SYM=SYMBOL(1)	JPT 1020
	GO TO 145	JPT 1030
C		JPT 1040
C	IF MULTIPLE DATA POINTS IN SAME PLOT LOCATION USE = SIGN	JPT 1050
C	***NOTE*** = SIGN IS INHIBITED FOR BAR GRAPH FORM OF OUTPUT	JPT 1060
C		JPT 1070
140	SYM=SYMBOL(1)	JPT 1080
145	PLOT(JY,JWORD)=JPUT(AMASK,JY,JWORD,JPOS,SYM,S!,PLOT)	JPT 1090
C		JPT 1100
C	GENERATE X AND Y SCALES	JPT 1110
C		JPT 1120
	DO 150 I=1,S1	JPT 1130
150	YSCALE(I)=FLOCAT(I-1)/YSC+YMIN	JPT 1140
	DO 155 I=1,I1	JPT 1150
	JANE=12-I	JPT 1160
	XLOW=XMIN-1.0/XSC	JPT 1170
	XINC=100.0/(XMAX-XLOW)	JPT 1180
	XUL1=(XMAX-FLOCAT(10*I-10)/XINC)	JPT 1190
	DMAS=ABS(XUL1)	JPT 1200
	IF (DMAS.LE.0.10) XUL1=0.0	JPT 1210
155	XSCALE(JANE)=XUL1	JPT 1220
	PRINT 160	JPT 1230
	DO 175 I=1,S1	JPT 1240
	JC=62-I	JPT 1250
	IT=11	JPT 1260
160	IT=IT-1	JPT 1270
	IF (IT.EQ.1) GO TO 165	JPT 1280
	IF (PLOT(JC,IT).EQ.DLANK) GO TO 150	JPT 1290
165	IF (N2.EQ.0) GO TO 170	JPT 1300
	PRINT 165, LB(I),YSCALE(JC),(PLOT(JC,J),J=1,IT)	JPT 1310
	GO TO 175	JPT 1320
170	PRINT 160, YSCALE(JC),(PLOT(JC,J),J=1,IT)	JPT 1330
175	CONTINUE	JPT 1340
	PRINT 160	JPT 1350
	PRINT 165	JPT 1360
	PRINT 200, (XSCALE(I),I=1,I1,2),(XSCALE(I),I=2,10,2)	JPT 1370

C	RETURN	JPT 1380
	180 FORMAT (13X,1H1,20(5H....I))	JPT 1390
	185 FORMAT (1X,A1,1X,E9.2,1X,1H+,10A10,AS)	JPT 1400
	190 FORMAT (2X,E10.3,1X,1H+,10A10,AS)	JPT 1410
	195 FORMAT (4X,11(9X,1HV))	JPT 1420
	200 FORMAT (4X,E10.3,3X,5(10X,E10.3)/7X,5(10X,E10.3))	JPT 1430
C	END	JPT 1440
	REAL FUNCTION JPUT(AMASK,J,JWORD,JPOS,SYM,NPDM,PLOT)	JPT 1450
C	JPUT ARRANGES DATA POINTS FOR PLOTTING	JPT 1460
	DIMENSION AMASK(10), PLOT(NPDM,10)	JP 10
	REAL JPUT	JP 20
	JPUT=(PLOT(J,JWORD).AND..NOT.AMASK(JPOS)).OR.(AMASK(JPOS).AND.SYM)	JP 30
	RETURN	JP 40
C	END	JP 50
	FUNCTION JLCOMP(I,K)	JP 60
C	JLCOMP ARRANGES DATA POINTS FOR PLOTTING	JP 70
	IF (I.GE.0.AND.K.LT.0) GO TO 105	JP 80
	IF (I.LT.0.AND.K.GE.0) GO TO 115	JP 90
	IF (I-K) 115,110,105	JP 100
105	JLCOMP=1	JLP 10
	RETURN	JLP 20
110	JLCOMP=0	JLP 30
	RETURN	JLP 40
115	JLCOMP=-1	JLP 50
	RETURN	JLP 60
C	END	JLP 70
	SUBROUTINE JSEP (ID1,ID2,M,LAB,LA)	JLP 80
C	JSEP SEPARATES THE ALPHABETS IN THE Y-AXIS LABEL FOR VERTICAL	JLP 90
C	***** DISPLAY	JLP 100
	DIMENSION LAB(ID1), LA(ID2)	JLP 110
	DATA LANK/10H	JLP 120
	IF (M.LE.0) GO TO 120	JLP 130
	DO 105 I=1,ID2	JLP 140
105	LA(I)=LANK	JLP 150
	LIM=(M-1)/10+1	JLP 160
	DO 115 I1=1,LIM	JLP 170
	N=11*I1-I1+1	JLP 180
	LABEL=LAB(I1)	JLP 190
	K=LABEL	JLP 200
	DO 115 I2=1,10	JSP 10
	JJ=N-I2	JSP 20
	IF (JJ.GT.M) GO TO 110	JSP 30
	K=MOD(K,64)	JSP 40
C	***** BOTH ISHFTLA AND ISHFTRA ARE SYSTEM DEPENDENT ROUTINES *****	JSP 50
C	K=ISHFTLA(K,54)	JSP 60
	LA(JJ)=K	JSP 70
110	LABEL=ISHFTRA(LABEL,6)	JSP 80
115	K=LABEL	JSP 90

```

120 RETURN                                JSP 270
C                                          JSP 280
C      END                                JSP 290
C      SUBROUTINE KFRINC (INTYP,NFREQ)      KFI 10
C                                          KFI 20
C*****                                KFI 30
C      *                                *KFI 40
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION: *KFI 50
C      *      1. COMPUTE THE FREQUENCY INCREMENTS FOR FREQUENCY SWEEP *KFI 60
C      *      CAPABILITY. *KFI 70
C      *                                *KFI 80
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES: *KFI 90
C      *      INTYP      : TYPE OF FREQUENCY INCREMENTS (LIN OR LOG) *KFI 100
C      *      NFREQ      : NUMBER OF INPUT FREQUENCIES (LE. 5) *KFI 110
C      *      ALL OTHER VARIABLE NAMES AND ARRAYS AS DEFINED IN *KFI 120
C      *      SUB-PROGRAM AGAIN. *KFI 130
C*****                                KFI 140
C                                          KFI 150
C      COMPLEX PHASE *KFI 160
C      COMMON /003/ FREQ(10),AMP(10),PHASE(10),LUNIT *KFI 170
C      COMMON /004/ NSTPS(5),FRINC(5),KFR(5) *KFI 180
C                                          KFI 190
C      *KFI 200
C***** CHECK IF LINEAR OR LOG INCREMENT IS DESIRED *KFI 210
C                                          KFI 220
C      IF (INTYP.EQ.3) GO TO 110 *KFI 230
C                                          KFI 240
C***** LOG FREQUENCY INCREMENTS *KFI 250
C                                          KFI 260
C      DO 105 I=1,NFREQ *KFI 270
C          STPS=FLOAT(NSTPS(I)-1) *KFI 280
C          105 FRINC(I)=(KFR(I)/FREQ(I))*((1.000/(STPS-1.000))) *KFI 290
C          RETURN *KFI 300
C                                          KFI 310
C***** LINEAR FREQUENCY INCREMENTS *KFI 320
C                                          KFI 330
C      110 DO 115 I=1,NFREQ *KFI 340
C          STPS=FLOAT(NSTPS(I)-1) *KFI 350
C          115 FRINC(I)=(KFR(I)-FREQ(I))/STPS *KFI 360
C          RETURN *KFI 370
C                                          KFI 380
C      END *KFI 390
C      SUBROUTINE KFRULS (I,NFREQ,IFLAG) *KFI 10
C                                          KFI 20
C*****                                KFI 30
C      *                                *KFI 40
C***** THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTION: *KFI 50
C      *      1. COMPUTE THE INCREMENTED FREQUENCY VALUES FOR THE NEXT *KFI 60
C      *      ANALYSIS. *KFI 70
C      *                                *KFI 80
C***** THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES: *KFI 90
C      *      I          : NUMBER OF INCREMENTS (ANALYSIS) CARRIED OUT *KFI 100
C      *      IFLAG      : (-1): NO FREQUENCY VALUES CHANGED *KFI 110
C      *      (0): FREQUENCY VALUES CHANGED *KFI 120
C      *      (1): FREQUENCY VALUES CHANGED *KFI 130
C      *      ALL OTHER VARIABLE NAMES AND ARRAYS AS DEFINED IN *KFI 140
C      *      SUB-PROGRAM AGAIN. *KFI 150
C*****                                KFI 160
C                                          KFI 170
C*****                                KFI 180

```

C	COMPLEX PHASE	KFU	190
	COMMON /003/ FR(10),AMP(10),PHASE(10),LUNIT	KFU	200
	COMMON /004/ NSTPS(5),FRINC(5),HFR(5)	KFU	210
	IFLAG=0	KFU	220
C		KFU	230
C*****	INCREMENT THE INPUT FREQUENCIES	KFU	240
C		KFU	250
	DO 105 J=1,NFREQ	KFU	260
	IF (I.GT.NSTPS(J)) GO TO 105	KFU	270
	FR(J)=FR(J)+FRINC(J)	KFU	280
	IFLAG=1	KFU	290
105	CONTINUE	KFU	300
	IF (IFLAG.EQ.0) RETURN	KFU	310
	I=I+1	KFU	320
C		KFU	330
C*****	PRINT THE NEW FREQUENCY VALUES	KFU	340
C		KFU	350
	WRITE (6,115) LUNIT	KFU	360
	DO 110 J=1,NFREQ	KFU	370
110	WRITE (6,120) J,FR(J)	KFU	380
	RETURN	KFU	390
C		KFU	400
	115 FORMAT (1H1,18HINPUT FREQUENCIES: ,/1H ,5HFREQUENCY,5X,6HVALUE(,A3,	KFU	410
	11H))	KFU	420
	120 FORMAT (1H0,4X,11,9X,E10.3)	KFU	430
C		KFU	440
	END	KFU	450
	SUBROUTINE LTRANS (NEG,N1,NADD,KK,NLBN,ER,NFROM,NT0,TYPE,ICONV,VAL	KFU	460
	LUE,NNODE,KEY)	LTR	10
C		LTR	20
C*****		LTR	30
C	* THIS SUB-PROGRAM PERFORMS THE FOLLOWING FUNCTIONS:	LTR	40
C	* 1. READ THE BIPOLAR TRANSISTOR PARAMETERS SPECIFIED BY	LTR	50
C	* THE USER.	LTR	60
C	* 2. CALCULATE THE COEFFICIENTS OF THE NONLINEAR ELEMENTS	LTR	70
C	* PRESENT IN THE EQUIVALENT TRANSISTOR MODEL.	LTR	80
C	* 3. FORM TOPOLOGY DESCRIPTION ARRAYS BASED ON THE	LTR	90
C	* EQUIVALENT REPRESENTATION.	LTR	100
C		LTR	110
C		LTR	120
C*****	THIS SUB-PROGRAM'S GLOSSARY OF FORTRAN NAMES:	LTR	130
C	* NEG : USER SPECIFIED ELEMENT(DEVICE) NUMBER	LTR	140
C	* N1 : NODE NUMBER FOR THE BASE TERMINAL	LTR	150
C	* NADD : CURRENT HIGHEST BRANCH NUMBER IN THE LINEAR	LTR	160
C	* NETWORK	LTR	170
C	* KK : UPON ENTRANCE: CURRENT NUMBER OF NONLINEAR	LTR	180
C	* ELEMENTS; UPON EXIT: NUMBER OF NONLINEAR	LTR	190
C	* ELEMENTS AFTER INCLUSION OF TRANSISTOR NON-	LTR	200
C	* LINEAR ELEMENTS	LTR	210
C	* ALL OTHER VARIABLE NAMES AND ARRAYS AS DEFINED IN	LTR	220
C	* SUB-PROGRAM AMAIN.	LTR	230
C		LTR	240
C*****		LTR	250
C		LTR	260
	INTEGER ER,TYPE,R,C	LTR	270
	REAL IE,IC,ICMAX,MD,M1,M2,M3,N,K,MU	LTR	280
	DIMENSION ER(1), NFROM(1), NTO(1), TYPE(1), VALUE(1), KEY(1), ICON	LTR	290
	IT(1), NLBN(1)	LTR	300
	COMMON /001/ NTYPE(10),A(10,9)	LTR	310
	COMMON /016/ NCONT(32),JCONT(10)	LTR	320

COMMON /ENDS/ NCAP,NDUS,NRES,NIND,NDCS,NCS	LTR 330
DATA R,C,NR,NC,ND/2H R,2H C,2HNR,2HNC,2HND/	LTR 340
C	LTR 350
C***** NODE NUMBERS FOR EMITTER,COLLECTOR,AND INTERNAL JUNCTION	LTR 360
C	LTR 370
NE=N1+3	LTR 380
NCJ=N1+2	LTR 390
NJ=N1+1	LTR 400
C	LTR 410
C***** READ TRANSISTOR PARAMETERS	LTR 420
C	LTR 430
READ (5,120) N,UCB,UCBD,MU,IC,ICMAX,AP,HFEMAX	LTR 440
READ (5,120) K,REF,CJE,CP2,RB,RC,C1,C3	LTR 450
C	LTR 460
C***** EMITTER RESISTIVE NONLINEARITY	LTR 470
C	LTR 480
NADD=NADD+1	LTR 490
ER(NADD)=NADD	LTR 500
NLEN(KK)=NADD	LTR 510
NFROM(NADD)=NJ	LTR 520
NTO(NADD)=NE	LTR 530
TYPE(NADD)=NR	LTR 540
HFE=HFEMAX/(1.00+AP*((ALOG10(IC/ICMAX))**2))	LTR 550
IE=IC*(1.00+1.00/HFE)	LTR 560
G1=37.9*IE	LTR 570
A(KK,1)=G1	LTR 580
A(KK,2)=G1**2/IE/2.00000	LTR 590
A(KK,3)=G1**3/IE**2/6.000000	LTR 600
C	LTR 610
C***** COLLECTOR DEPENDENT NONLINEARITY	LTR 620
C	LTR 630
M0=1.00/(1.00-(UCB/UCBD)**N)	LTR 640
M1=N*UCB**(N-1)*M0**2/UCBD**N	LTR 650
M2=(N-1.0000)*M1/UCB/2.00+M1**2/M0	LTR 660
DUM1=2.00*M2*((N-1.0000)/2.00/UCB+2.0*M1/M0)/3.0000	LTR 670
DUM2=M1*((N-1.0000)/2.00/UCB**2+(M1/M0)**2)/3.000	LTR 680
M3=DUM1-DUM2	LTR 690
SM1=IC*M1/M0	LTR 700
SM2=IC*M2/M0	LTR 710
SM3=IC*M3/M0	LTR 720
DUM2=ALOG10(2.718281828)*2.000*AP	LTR 730
DUM1=ALOG10(IC/ICMAX)	LTR 740
A1=HFEMAX/(HFEMAX+1.00+AP*DUM1**2+DUM1*DUM2)	LTR 750
A2=-A1**3*DUM2*(DUM1+ALOG10(2.718281818))/2.0/IC/HFEMAX	LTR 760
A3=(A1/6.00)*(-2.00*A2/IC+12.00*(A2/A1)**2-A1**3*DUM2**2/2.00/AP/IE	LTR 770
IC**2/HFEMAX)	LTR 780
JJ=KK+1	LTR 790
NADD=NADD+1	LTR 800
NLEN(JJ)=NADD	LTR 810
ER(NADD)=NADD	LTR 820
NFROM(NADD)=NCJ	LTR 830
NTO(NADD)=NJ	LTR 840
TYPE(NADD)=ND	LTR 850
ICONT(NADD)=NADD+2	LTR 860
JCONT(JJ)=NADD+1	LTR 870
A(JJ,1)=A1*M0*A(KK,1)	LTR 880
A(JJ,2)=SM1	LTR 890
A(JJ,3)=A2*M0*A(KK,1)**2+A1*M0*A(KK,2)	LTR 900
A(JJ,4)=SM2	LTR 910
A(JJ,5)=A1*M1*A(KK,1)	LTR 920

A(JJ,6)=A3*M0*A(KK,1)**3+A1*M0*A(KK,3)+2.0*A2*M0*A(KK,1)*A(KK,2)	LTR 930
A(JJ,7)=SM3	LTR 940
A(JJ,8)=A2*M1*A(KK,1)**2+A1*M1*A(KK,2)	LTR 950
A(JJ,9)=A1*M2*A(KK,1)	LTR 960
C	LTR 970
C***** COLLECTOR-BASE CAPACITANCE	LTR 980
C	LTR 990
NADD=NADD+1	LTR 1000
BR(NADD)=NADD	LTR 1010
NFROM(NADD)=NCJ	LTR 1020
NTD(NADD)=N1	LTR 1030
IF (ABS(C3).EQ.0.0000000) GO TO 105	LTR 1040
TYPE(NADD)=C	LTR 1050
VALUE(NADD)=C3	LTR 1060
KEY(NADD)=2	LTR 1070
NCAP=NCAP+1	LTR 1080
GO TO 110	LTR 1090
105 VALUE(NADD)=1.000E+06	LTR 1100
TYPE(NADD)=R	LTR 1110
KEY(NADD)=5	LTR 1120
NRES=NRES+1	LTR 1130
C	LTR 1140
C***** EMITTER CAPACITOR(LINEAR)	LTR 1150
C	LTR 1160
110 NADD=NADD+1	LTR 1170
BR(NADD)=NADD	LTR 1180
NFROM(NADD)=NJ	LTR 1190
NTD(NADD)=NE	LTR 1200
TYPE(NADD)=C	LTR 1210
VALUE(NADD)=CJE+IE*CP2	LTR 1220
KEY(NADD)=2	LTR 1230
NCAP=NCAP+1	LTR 1240
C	LTR 1250
C***** BASE-EMITTER CAPACITANCE(LINEAR)	LTR 1260
C	LTR 1270
IF (ABS(C1).EQ.0.000) GO TO 115	LTR 1280
NADD=NADD+1	LTR 1290
BR(NADD)=NADD	LTR 1300
NFROM(NADD)=NJ	LTR 1310
NTD(NADD)=NE	LTR 1320
TYPE(NADD)=C	LTR 1330
VALUE(NADD)=C1	LTR 1340
KEY(NADD)=2	LTR 1350
NCAP=NCAP+1	LTR 1360
C	LTR 1370
C***** COLLECTOR CAPACITIVE NONLINEARITY	LTR 1380
C	LTR 1390
115 LL=KK+2	LTR 1400
NADD=NADD+1	LTR 1410
BR(NADD)=NADD	LTR 1420
NFROM(NADD)=NCJ	LTR 1430
NTD(NADD)=NJ	LTR 1440
TYPE(NADD)=NC	LTR 1450
NLBN(LL)=NADD	LTR 1460
A(LL,1)=K/UCB**MU	LTR 1470
A(LL,2)=-A(LL,1)/UCB/6.000	LTR 1480
A(LL,3)=A(LL,1)/UCB**2/27.00	LTR 1490
C	LTR 1500
C***** COLLECTOR RESISTANCE(LINEAR)	LTR 1510
C	LTR 1520

5-4. System Dependent Cards

Program PRANC was developed on the CDC 6500/6600 computer system at Purdue University. The system dependent cards contained in the program are listed in Table 5-1.

Table 5-1. System Dependent Cards

Sub-Program	Card Identification Number
AMAIN	AMN 2200,6000,6430
GZOC	GZC 380
JSEP	JSP 230,250

The sub-programs, and their functions called by the cards listed in Table 5-1 are as follows:

SECOND: Subroutine SECOND is used to determine the elapsed time in seconds in performing a sequence of PRANC phrases.

LINEQ4: Subroutine LINEQ4 is a linear equation solver routine, used to invert a complex matrix.

ISHFTLA (I,N): is used to perform an N-place arithmetic left shift on I (circular).

ISHFTRA (I,N): is used to perform an N-place arithmetic right shift on I (end-off, sign fill); e.g. K = ISHFTRA (1,1) sets K to 0; K = ISHFTRA (1,0) sets K to 1.

CHAPTER 6
CONCLUDING REMARKS

As stated earlier, the fundamental objective underlying this research effort was to examine the computational aspect of the Volterra series method. In the process, we developed an efficient algorithm for adapting the Volterra series method for computer-aided analysis of nonlinear circuits. A semi-symbolic approach for analyzing the linearized part of the nonlinear circuit was used as the basis for this development. The algorithm was implemented in a computer program, entitled PRANC. The main contributions of this effort may thus be identified as follows:

- (1) The development of an efficient algorithm for adapting the Volterra series method for computer-aided analysis.
- (2) The development of a symbolic approach for analyzing the linearized circuit.
- (3) The development of a digital computer program for the spectrum analysis of nonlinear circuits.

As part of the effort, several network examples were exercised on PRANC. The execution times involved in these examples indicate that PRANC is highly efficient from a computational standpoint. Networks with several nonlinearities, several energy storage elements (as in Example 4-2), and multiple input frequencies involve execution times which are small and easily affordable.

The fundamental criterion in the development of PRANC was computational efficiency. The results from the use of PRANC indicate that this criterion

has been met successfully. The "ease of use", which is another important performance measure in software development, was not given as much weight in this effort. As part of continuing work, it is recommended that several user-oriented features, such as free-format input, built-in device modeling, parameter variation feature, etc., be incorporated in the program. The computational efficiency inherent in the present version of PRANC together with certain "ease of use" features should render it a powerful tool for analyzing nonlinear circuits.

REFERENCES

- [1] Volterra, V., Theory of Functionals and of Integral and Integro-Differential Equations, New York: Dover, 1959.
- [2] Wiener, N., Nonlinear Problems in Random Theory, New York: Technology Press, 1958.
- [3] James, Z. D., "Nonlinear Operators for System Analysis," MIT Research Lab. of Electronics, Tech. Rept. 370, Aug. 25, 1960.
- [4] Brilliant, M. B., "Theory of the Analysis of nonlinear systems," MIT Research Lab. of Electronics, Tech. Rept. 345, March 3, 1958.
- [5] George, D. A., "Continuous Nonlinear Systems," MIT Research Lab. of Electronics, Tech. Rept. 355, July 24, 1959.
- [6] Barrett, J. F., "The Use of Functionals in the Analysis of Nonlinear Physical Systems," Part-I, Sect. 2, A. T. Fuller, ed., Nonlinear Stochastic Control Systems, London: Taylor and Francis, 1970.
- [7] Narayanan, S., "Transistor Distortion Analysis Using Volterra Series Representation," BSTJ, 46, pp. 991-1023, May-June 1967.
- [8] Van Trees, H., Synthesis of Optimum Nonlinear Control, Mass.: MIT Press, 1962.
- [9] Van Trees, H., "Functional Techniques for the Analysis of the Nonlinear Behavior of Phase Locked Loops," Proc. IEEE, Vol. 52, No. 8, pp. 894-911, Aug. 1964.
- [10] Graham, J. and Ehrman, L., eds., Nonlinear System Modeling and Analysis with Application to Communication Receivers, Rome Air Development Center, RADC-TR-73-178 (AD-766278), Griffis AFB, June 1973.
- [11] Bedrosian, E. and Rice, S., "The Output Properties of Volterra Systems (Nonlinear Systems with Memory) Driven by Harmonic and Gaussian Inputs," Proc. IEEE, Vol. 59, No. 12, pp. 1688-1707, Dec. 1971.
- [12] Bussgang, J. J., et. al., "Analysis of Nonlinear Systems with Multiple Inputs," Proc. IEEE, Vol. 62, No. 8, pp. 1088-1119, Aug. 1974.
- [13] Thapar, H. K. and B. J. Leon, "Lumped Nonlinear System Analysis with Volterra Series," School of Electrical Engineering, Purdue University, TR-EE-32, July 1979.
- [14] Leon, B. J., and Schaefer, D. J., "Volterra Series and Picard Iteration for Nonlinear Circuits and Systems," IEEE Trans., CAS-25, No. 9, pp. 789-793, Sept. 1978.

- [15] Lubbock, J. K., and Bansal, V. S., "Multidimensional Laplace Transforms for Solution of Nonlinear Equations," in Multidimensional Systems: Theory and Applications, N. K. Bose, ed., IEEE Press, pp. 155-162, 1979.
- [16] Down, T., "Symbolic Evaluation of Transmittances from the Nodal Admittance Matrix," Elec. Letters, Vol. 5, No. 16, pp. 379-380, Aug. 1969.
- [17] -, "Inversion of the Nodal-Admittance Matrix in Symbolic Form," Elect. Letters, Vol. 6, No. 3, pp. 74-76, Feb. 1970.
- [18] -, "Inversion of the Nodal Admittance Matrix for Active Networks in Symbolic Form," Elect. Letters, Vol. 6, No. 22, pp. 690-691, Oct. 1970.
- [19] Lin, P. M., "Formulation of Hybrid Matrices for Linear Multiports Containing Controlled Sources," IEEE Trans. Vol. CT-21, pp. 169-175, March 1974.
- [20] Chua, L. O., and Lin, P. M., Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques, Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [21] Dahlquist, G., Bjork, A., Numerical Methods, NJ: Prentice Hall, 1974.
- [22] Feller, W., An Introduction to Probability Theory and Its Applications, John Wiley and Sons, NY, 1957.
- [23] Thapar, H. K. and Leon, B. J., "Computer-Aided Distortion and Spectrum Analysis of Nonlinear Circuits," IEEE Int'l. Symposium on CAS, Tokyo, pp. 46-49, July 1979.
- [24] "NCAP: Nonlinear Circuit Analysis Program," Rome Air Development Center, Griffis AFB, NYC, Private Communication.
- [25] Lin, P. M., "A Survey of Applications of Symbolic Network Functions," IEEE Trans., Vol. CT-20, pp. 732-737, Nov. 1973.
- [26] McClenahan, J., and Chan, S. P., "Computer Analysis of General Linear Networks Using Digraphs," Internl. J. Electronics, No. 22, pp. 153-191, 1972.
- [27] Singhal, K., and Vlach, J., "Symbolic Analysis of Analog and Digital Circuits," IEEE Trans., Vol. CAS-24, No. 11, pp. 598-609, Nov. 1977.
- [28] Parlett, B. N., "The LU and QR Algorithms," in A. Ralston and H. S. Wilf, eds., Numerical Methods for Digital Computers, NYC: John Wiley and Sons, Inc., 1967, Vol. II, pp. 116-130.
- [29] Acton, F. S., Numerical Methods that Work, Harper & Row, Publishers, NYC, 1970.

Appendix A. A DEVICE MODELLING EXAMPLE

In this section we present an example of how to obtain mathematical models for nonlinear devices. The mathematical models so developed can then be used to obtain equivalent circuits for analysis purposes.

Most devices commonly encountered in electronics, where one would be interested in computing the harmonic distortion due to the nonlinear operation, are operated in the active region where the device operation is quasi-linear about an operating point established by the circuit bias. Here we develop the incremental model for some such devices. It is important to make a distinction between total and incremental nonlinear circuits. Total model, or global models, interrelate the total instantaneous voltages, current, and/or charges in the device. Such models are used for operating point or large-signal analysis. The incremental or small-signal models for devices are derived from these global models by some kind of an approximation (usually a Taylor series expansion) around the operating point. In deriving incremental models, it is desirable to have a model that is independent of the bias point in the normal active region, so that the nonlinear effects due to a change in the operating point can be predicted.

We now present a mathematical model for a semiconductor diode. In the commonly used small-signal applications of semiconductor diodes, two types of operations are encountered: (1) forward-bias (e.g. mixers); (2) reverse-bias (varactor converter).

In the forward-bias operation, the primary nonlinearity is a memoryless nonlinearity given by

$$I = I_s [\exp(qV/nkT) - 1] \quad (1)$$

where n is the ideality factor for the diode. Then for the forward-biased diode with a small-signal input, we can write eqn. (1) as:

$$I_D + i_d = I_s [\exp(qV_D/nkT) \exp(qv_d/nkT) - 1] \quad (2)$$

where I_D and V_D are the bias current and voltage, respectively, and i_d and v_d are the incremental current and voltage. For $(qv_d/nkT) < 1$, we have a convergent Taylor series for:

$$\exp(qv_d/nkT) = \sum_{s=0}^{\infty} \frac{1}{s!} \left(\frac{qv_d}{nkT} \right)^s \quad (3)$$

Substituting (3) into (2) and approximating

$$I_D = I_s \left[\exp \frac{qV_D}{nkT} - 1 \right] \approx I_s \exp \frac{qV_D}{nkT},$$

we obtain the following:

$$i_d = I_D \frac{q}{nkT} v_d + \frac{I_D}{2!} \frac{q}{nkT}^2 v_d^2 + \frac{I_D}{3!} \frac{q}{nkT}^3 v_d^3 + \dots \quad (4)$$

which is in a form suitable for analysis on PRANC.

In the case of the reverse-biased diode, the primary nonlinearity is the nonlinear junction capacitance $C(V)$, where $C(V)$ is of the following form:

$$C(V) = \frac{C(0)}{[1 - V/\phi]^k} \quad (5)$$

where $C(0)$, ϕ , and k are generally specified by the manufacturer.

The charge stored in the capacitor of eqn. (5) is:

$$Q(V) = \int_0^V C(v) dv$$

$$= \frac{\phi}{(k-1)} \frac{C(0)}{[1 - V/\phi]^{(k-1)}} \quad (6)$$

Expanding eqn. (6) into a Taylor series yields:

$$Q(V_c + v_c) = C(V_c) \left[\frac{(\phi - V_c)}{(k-1)} + v_c + \frac{kv_c^2}{2!(\phi - V_c)} + \frac{k(k+1)v_c^3}{3!(\phi - V_c)^2} + \dots \right] \quad (7)$$

The incremental capacitor current is the change of total charge with respect to time. Since V_c is a constant, we get

$$i_c = \frac{dQ}{dt} = C(V_c) \frac{dv_c}{dt} + \frac{kC(V_c)}{2!(\phi - V_c)} \frac{dv_c^2}{dt} + \frac{k(k+1)C(V_c)}{3!(\phi - V_c)^2} \frac{d}{dt} v_c^3 + \dots \quad (8)$$

Equation (8) is the mathematical model of the incremental nonlinear capacitance current. The first term is a linear capacitor of value $C(V_c)$, and the term in v_c^n represent the nonlinear capacitive terms. Again, note that eqn. (8) is in a form suitable for analysis on PRANC.

The models for other nonlinear devices, such as transistors, JFETS, vacuum tubes, etc., can be found using the same kind of an approach.



MISSION of Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C³I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.